ANSI/ISA-S50.02-1997, Part 4
Approved January 6, 1998

Standard

Fieldbus Standard for Use in Industrial Control Systems Part 4: Data Link Protocol Specification

S50.02, Fieldbus Standard for Use in Industrial Control Systems, Part 4: Data Link Protocol Specification

ISBN: 1-55617-642-2

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

Preface

This preface, as well as all footnotes and annexes F-H, is included for informational purposes and is not part of ISA-S50.02, Part 4.

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

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

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

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

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

ADDITIONALLY, ISA HAS BEEN INFORMED THAT THIS STANDARD CONTAINS A PATENTED TECHNOLOGY OWNED BY YAMATAKE-HONEYWELL CO., LTD., TOKYO, JAPAN. YAMATAKE-HONEYWELL HAS GRANTED NON-EXCLUSIVE, ROYALTY-FREE USE OF THIS PATENT AS LONG AS IT IS USED TO IMPLEMENT THIS PART OF THIS ISA STANDARD. THE DETAILED DOCUMENT IS FILED IN THE ISA OFFICE. ALL OTHER RIGHTS TO THE PATENT ARE RESERVED BY YAMATAKE-HONEYWELL. ISA MAKES NO REPRESENTATION AS TO THE REASONABLENESS, VALIDITY, OR APPLICATION OF THE LICENSE OFFERED BY

YAMATAKE-HONEYWELL. FOR FURTHER INFORMATION, PLEASE CONTACT: INTELLECTUAL PROPERTY DEPARTMENT, YAMATAKE-HONEYWELL CO., LTD., SHIBUYAKU SHIBUYA 2-12-19, TOKYO 150, JAPAN.

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

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

* One vote per company

The following people served as members of ISA Committee SP50:

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

M. Dajani	Philip Morris USA
E. Delahostria	Allen-Bradley
U. Doebrich	Siemens AG
J. Dohoney	Jet Propulsion Lab
B. Dumortier	Schneider Electric
*J. Elias	Honeywell, Inc.
T. Feindel	R. Stahl, Inc.
M. Giraudi	Elsag Bailey
J. Gray	Krohne America Inc.
*P. Griem, Jr.	Honeywell, Inc.
B. Gross	General Motors Corporation
*C. Gross	Dow Chemical Company
A. Gupta	I/OLink, Inc.
V. Jacobson	Ronan Engineering Company
J. Jamison	Kenonic Controls Limited
K. Kakizakai	Fuji Electric Company, Ltd.
*R. Ketcham	Union Carbide Corporation
T. Kuusisto	Valmet Automation Inc.
*T. Laine	Cegelec
H. Landet	Fieldbus International A/s FINT
C. Langford	Cullen G. Langford Inc.
*R. Lasher	Bailey Controls Company
A. Lesh	AMP, Inc.
P. Leviti	Marconi Italiana Automazione
K. Lindner	Endress + Hauser GmbH + Company
T. Madden	Exxon Company USA
A. McCauley, Jr.	Chagrin Valley Controls, Inc.
F. McKenna	FMcK Associates, Ltd.
R. Meeker	Process Control Solutions, Inc.
D. Merriman	MerTech, Inc.
E. Messano	Valtek International
O. Mirabella	Universita Degli Studi di Catania
D. Modell	MPI
A. Mukherji	SAI
*G. Nachev	IsoMatic Laboratories
L. Natiello	Merck & Company Inc.
L. Neitzel	Fieldbus Foundation
H. Newman	Cornell University
*P. Noury	Cegelec
*E. Pageler	Fisher•Rosemount Systems Inc.
P. Patel	V Automation
*T. Phinney	Honeywell Inc.
F. Russo	ENEL-DSR
K. Schwarz	Schwarz Communication Consult
*E. Sederlund	Dow Chemical Company
Y. Shimanuki	Toshiba Corporation
E. Skabowski	Tendent Consulting
J. Slavinsky	Czech Standards Institute
D. Sniezek	Lockheed Martin Federal Services
J. Sprague	Saudi Aramco Oil Company
R. Squires	Measurement Technology Limited
H. Storey	Shell Oil Company
R. Szanyi	Mobil Technology Company
N. Tobol	Ronan Engineering Company
*A. Tresset	Cegelec
C. Vaidya	Sun Microsystems
C. Van Haren	James River Corporation

J. Weiss	Electric Power Research Institute
C. Williams	Eastman Kodak Company
G. Winchester	National Electrical Manufacturers Association
*D. Wroblewski	Bailey Controls Company
*M. Zielinski	Fisher•Rosemount Systems Inc.

This published standard was approved for publication by the ISA Standards and Practices Board on August 13, 1997.

NAME	COMPANY
R. Webb, Vice President	Pacific Gas & Electric Company
H. Baumann	H. D. Baumann Associates Limited
D. Bishop	Chevron USA Production Company
P. Brett	Honeywell, Inc.
W. Calder III	Factory Mutual Research Corporation
M. Cohen	Flexonics, Inc.
H. Dammeyer	Ohio State University
R. Dieck	Pratt & Whitney
W. Holland	Southern Company Services, Inc.
H. Hopkins	Utility Products of Arizona
A. Iverson	Ivy Optiks
K. Lindner	Endress + Hauser GmbH & Company
V. Maggioli	Feltronics, Inc.
T. McAvinew	Instrumentation & Control Engineering LLC
A. McCauley, Jr.	Chagrin Valley Controls, Inc.
G. McFarland	Honeywell Indus. Auto. & Cont.
E. Montgomery	Fluor Daniel, Inc.
D. Rapley	Rapley Engineering Services Inc.
R. Reimer	Rockwell Automation A-B
J. Rennie	Factory Mutual Research Corporation
W. Weidman	Consultant
J. Weiss	Electric Power Research Institute
J. Whetstone	National Institute of Standards & Technology
M. Widmeyer	Carnegie-Mellon University
H. Wiegle	Canus Corporation
C. Williams	Eastman Kodak Company
G. Wood	Graeme Wood Consulting
M. Zielinski	Fisher•Rosemount Systems Inc.

* One vote per company

Contents

Figures

Tables

Introduction

Keywords

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

General

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

Purpose

The purpose of this part of this International Standard is to define the Fieldbus Data Link Protocol. It is most closely related to, and lies within the field of application of, the Fieldbus Data Link Service Definition.

The Data Link Protocol provides the Data Link Service by making use of the services available from the Physical Layer. The relationship between the International Standards for Fieldbus Data Link Service, Fieldbus Data Link Protocol, Fieldbus Physical Service, and OSI Network or Fieldbus Application Protocol is illustrated in Figure 1.

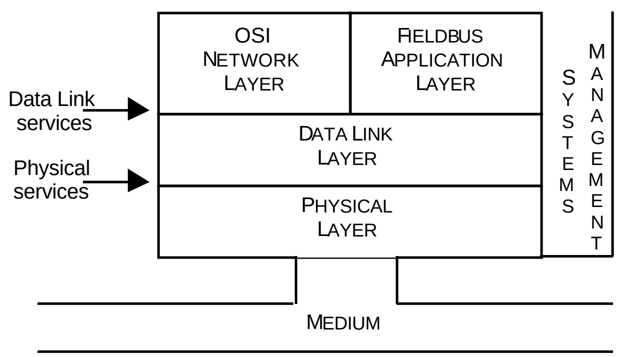


Figure 1 — Relationship of this part of this International Standard to other Fieldbus and OSI standards

Intended Users

The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer Data Link entities 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; and

d) as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard, together with other standards positioned within the OSI or Fieldbus Reference Models, otherwise incompatible systems may work together in any combination.

1 Scope

The Fieldbus Data Link protocol specification is a Data Link Layer Standard designed to support time-critical messaging communications to and from devices in an automation environment.

1.1 Specifications

This standard specifies

a) procedures for a single protocol 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; and

b) the structure of the Fieldbus Data Link (DL) Protocol Data Units used for the transfer of data and control information, and their representation as Physical Interface Data Units.

1.2 Procedures

The procedures are defined in terms of

a) the interactions between peer DL-entities (DLEs) through the exchange of Fieldbus Data Link Protocol Data Units;

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

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

1.3 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. Clause 10 specifies profiles for this Fieldbus protocol.

1.4 Conformance

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

The supplier of a protocol implementation which is claimed to conform to this International Standard shall complete a copy of the PICS Proforma (see Annex E), and shall provide the information necessary to identify both the supplier and the implementation.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

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

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

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

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

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

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

ISO/IEC 9646-1 : 1991, *Information technology — Open systems interconnection — Conformance testing methodology and framework — Part 1: General concepts.*

ISO/IEC 9646-2 : 1991, *Information technology — Open systems interconnection — Conformance testing methodology and framework — Part 2: Abstract test suite specification.*

ISO/IEC TR 18802 : 1993, *Information technology — Telecommunications and information exchange between systems — Overview of Local Area Network Standards.*

ISO/IEC 10038 : 1993, *Information technology — Telecommunications and information exchange between systems — Local area networks — Media access control (MAC) bridges.*

ISA S50.02-2 : 1993, *Digital data communications for measurement and control — Fieldbus for use in industrial control systems — Part 2 : Physical layer specification and service definition.*

[ISA S50.02-3], *Digital data communications for measurement and control — Fieldbus for use in industrial control systems — Part 3 : Data link service definition.*

3 Definitions

For the purposes of this part of this International Standard, the following definitions apply.

3.1 Reference model definitions

This part of this International Standard is based in part on the concepts developed in ISO 7498 parts 1 and 3, and in ISO 7498/AD1, and makes use of the following terms defined therein:

3.1.1	DL-address	[7498-3]
3.1.2	DL-address-mapping	[7498-1]
3.1.3	called-DL-address	[7498-3]
3.1.4	calling-DL-address	[7498-3]
3.1.5	centralized multi-end-point-connection	[7498-1]
3.1.6	DL-connection	[7498/AD1]
3.1.7	DL-connection-end-point	[7498-1]
3.1.8	DL-connection-end-point-identifier	[7498-3]
3.1.9	DL-connection-mode transmission	[7498/AD1]
3.1.10	DL-connectionless-mode transmission	[7498/AD1]
3.1.11	correspondent (N)-entities correspondent DL-entities correspondent Ph-entities	[7498-1]
3.1.12	DL-data-link	[7498-1]
3.1.13	DL-data-source	[7498-1]
3.1.14	demultiplexing	[7498-1]
3.1.15	DL-duplex-transmission	[7498-1]
3.1.16	(N)-entity DL-entity Ph-entity	[7498-1]
3.1.17	flow control	[7498-1]
3.1.18	Ph-interface-control-information	[7498-1]
3.1.19	Ph-interface-data	[7498-1]

3.1.20	Ph-interface-data-unit	[7498-1]
3.1.21	(N)-layer DL-layer (N=2) Ph-layer (N=1)	[7498-1]
3.1.22	layer-management	[7498-1]
3.1.23	multiplexing	[7498-1]
3.1.24	DL-name	[7498-3]
3.1.25	naming-(addressing)-authority	[7498-3]
3.1.26	naming-(addressing)-domain	[7498-3]
3.1.27	naming-(addressing)-subdomain	[7498-3]
3.1.28	peer-entities	[7498-1]
3.1.29	primitive name	[7498-3]
3.1.30	DL-protocol	[7498-1]
3.1.31	DL-protocol-connection-identifier	[7498-1]
3.1.32	DL-protocol-control-information	[7498-1]
3.1.33	DL-protocol-data-unit	[7498-1]
3.1.34	DL-protocol-version-identifier	[7498-1]
3.1.35	reassembling	[7498-1]
3.1.36	recombining	[7498-1]
3.1.37	DL-relay	[7498-1]
3.1.38	reset	[7498-1]
3.1.39	responding-DL-address	[7498-3]
3.1.40	routing	[7498-1]
3.1.41	segmenting	[7498-1]
3.1.42	sequencing	[7498-1]
3.1.43	(N)-service DL-service Ph-service	[7498-1]

3.1.44 (N)-service-access-point [7498-1]
DL-service-access-point
Ph-service-access-point

3.1.45 (N)-service-access-point-address [7498-3]
DL-service-access-point-address
Ph-service-access-point-address

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

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

3.1.48 DL-simplex-transmission [7498-1]

3.1.49 splitting [7498-1]

3.1.50 synonymous name [7498-3]

3.1.51 systems-management [7498-1]

3.1.52 DL-user-data [7498-1]

3.2 Service convention definitions

This part of this International Standard also makes use of the following terms defined in ISO/TR 8509 and ISO/IEC 10731 as they apply to the Data Link Layer:

3.2.1 acceptor

3.2.2 asymmetrical service

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

3.2.4 deliver (primitive)

3.2.5 DL-confirmed-facility

3.2.6 DL-facility

3.2.7 DL-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 Data Link Service definitions

For the purpose of this part of this International Standard, the following definitions also apply:

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

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

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

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

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

a) connect two or more separate DL-subnetworks (links) to form a unified DL-subnetwork (the extended link);

b) provide a means by which two end systems can communicate, when at least one of the end systems is periodically inattentive to the interconnecting DL-subnetwork; and

c) connect the extended link to a separate time-distribution network (for example, a national radio time-distribution network).

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

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

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

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

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

NOTES —

1. In this last case, the characterizations as publisher and subscriber are misnomers.

2. The publishing DLS-user should use rate control because the subscribing DLS-users cannot exert flow control on their publishing peer-entity. Similar considerations apply to the subscribing DLS-users.

3.3.9 publisher: The DL-entity of the DLS-user at the centralized end-point of a multi-peer connection. This DLS-user is also known as the **publishing** DLS-user; it can send only to the subscribing DLS-users as a group, but not to any smaller subset of the subscribers.

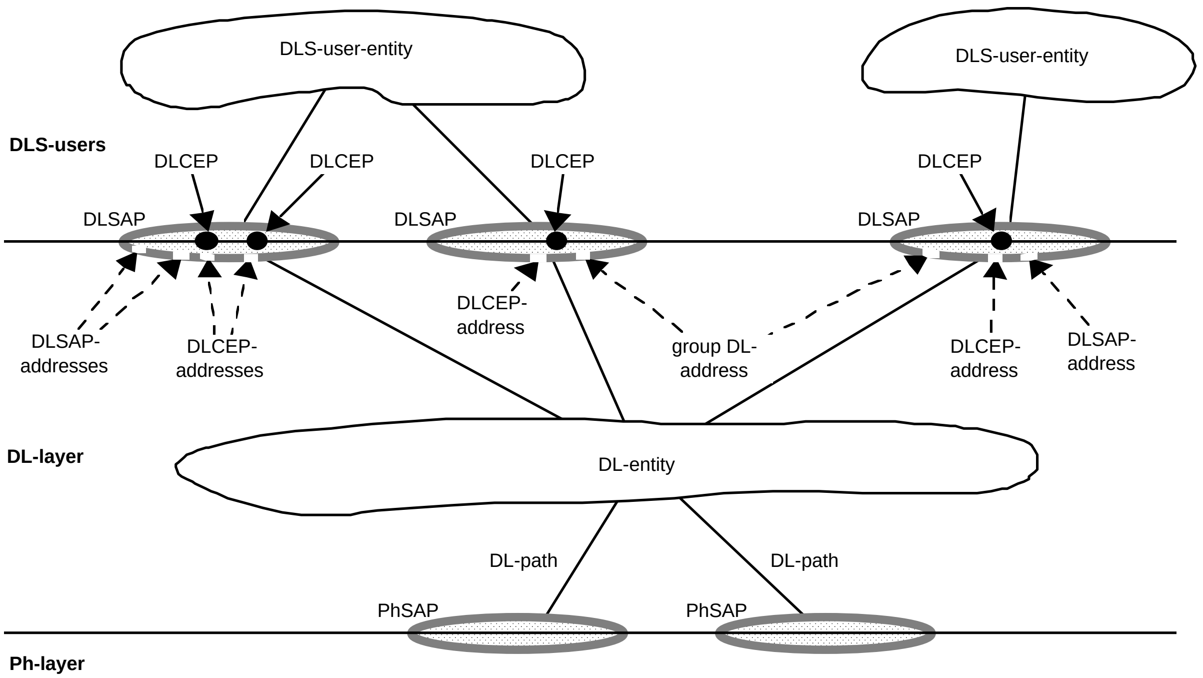
3.3.10 subscriber: The DL-entity of a non-publishing DLS-user of a multi-peer connection. Such a DLS-user is also known as a **subscribing** DLS-user; it can send only to the publishing DLS-user.

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

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

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

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



LEGEND

1. DLSAPs and PhSAPs are depicted as ovals spanning the boundary between two adjacent layers.

2. DL-addresses are depicted as designating small gaps (points of access) in the DLL portion of a DLSAP. A DLCEP-address also designates a specific point of infomraiton flow (its DLCEP) within the DLSAP.

3. A single DL-entity may have multiplied DLSAP-addresses and group DL-addresses assocaited with a single DLSAP.

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

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

3.3.14 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 of one or more DLS-users.

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

3.3.15 DLCEP-address: A DL-address which designates either

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

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

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

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

3.3.16 DLSEP-address: A DL-address which designates a DL-scheduling-end-point within a DLE.

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

3.3.17 NODE DL-address: A DL-address which designates the (single) DL-entity associated with a single node (see 3.3.3) on a specific local link.

3.3.18 transaction: Either

a) a sequence of two related DLPDUs which appear consecutively on the local link, where the first DLPDU required the immediate transmission of the second DLPDU; or

b) a single DLPDU on the local link which does not require the immediate transmission of a second DLPDU.

The DLE sending the first DLPDU of the transaction is known as the **initiator**; the DLE which sends the second DLPDU of the transaction, if any, is known as the **responder**.

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

3.3.19 immediate reply: A DLPDU sent by the responder in a transaction as a solicited reply to the immediately-prior DLPDU. The original DLPDU sent by the initiator of the transaction specifies the constraints on the replying DLPDU.

3.3.20 Link Active Scheduler (LAS): A special-purpose elective role of a DLE which schedules the local link and serves as the local source of DL-time for the link. The LAS functions exist within each link master (LM) DLE, which contends for and then activates those LAS functions after detecting the absence on the link of a DLE whose LAS functions are active. The DLE serving as LAS, sometimes referred to as the **LAS DLE**, receives, and responds to, scheduling requests from all DLEs on the link, including itself. It also receives and responds to requests for the current DL-time.

3.3.21 Link Master (LM): A DLE which can also provide the LAS functions for the link, including initializing and scheduling the link.

3.3.22 token: A right to transmit on the local link. This right is assumed by a DLE when it activates its LAS functions. This right may be delegated to individual DLEs, subject to specified constraints on its usage. In all cases, this right ultimately reverts to the DLE which has activated its LAS functions (the LAS DLE) .

Each token is implicitly qualified by the method of its transmission or assumption:

a) A **scheduler token** is assumed and held by the LAS DLE, and can be sent to another LM DLE on the local link to transfer the activation of LAS functions to that receiving DLE.

b) A **delegated token** is created by the LAS DLE and sent to a DLE on the local link, and is returned upon completion of its use, or assumed by the LAS DLE at its expiration.

c) A **reply token** is created by the current delegated token holder (or scheduler token holder if there is no delegated token holder) and sent to a DLE on the local link, requesting an immediate reply; it is returned with that immediate reply, or assumed by the current delegated token holder at the expiration of the reply period.

NOTE — It is possible, though not absolutely required, for a DLE to function simultaneously as an LAS, an initiator, and a responder, delegating a token and receiving that token, requesting an immediate reply and receiving that request, and then replying to itself as requestor and returning the token to itself as LAS. This would require the transmission of at least three DLPDUs — one from each role.

3.3.23 dominant token: The unique right to initiate the next transmission on the local link:

a) A reply token is the dominant token on the local link during the period after its creation and before its expiration or return, and the reply token holder (the responder) holds the unique right to transmit.

b) Otherwise, when no reply token exists, a delegated token is the dominant token on the local link during the period after its creation and before its expiration or return, and its token holder (the initiator) holds the unique right to transmit.

c) At all other times the scheduler token is the dominant token on the local link, and the LAS DLE (which functions as the initiator) holds the unique right to transmit.

3.3.24 node-timer: A frequency-adjustable counter, maintained by the local DLE, which is used to provide the DLS-user with a local multi-partite sense of DL-time (see 5.5.4 and 9.4.1.1) such that one component of that DL-time forms a monotonically-increasing sense of local time available for use within the attached end system. Conceptually, the node-timer counts in nominal units of 2^{-13} ms and has a period of over 100 years. Therefore, any actual counter shall be a binary counter whose least significant bit has a nominal weight of $2^{\pm N}$ ms, with a rollover period greater than the expected maximum interval between resets of the DLE.

NOTE 1 — When no information about that maximum interval is available, an interval of 5 yrs shall be assumed.

The node-timer is also used within the DL-protocol to provide a shared sense of DL-time which is used both to synchronize DLE scheduling actions, where appropriate, and to synchronize the rate of drift of all of the node-timers on the extended link. This latter is achieved by adjusting the frequency of each DLE's node-timer such that the DLE's monotonic sense of local time maintains an approximately constant phase relationship with that of the DLE serving as the **time-reference DLE**.

NOTE 2 — This adjustment is the reason why the weight of the node-timer unit of counting is only nominally $2^{\pm N}$ ms.

3.3.25 slot-time: A configuration parameter of each local link, measured as an **integral** multiple of the transmission period of one complete octet, with an integral value of between 1 and 4095 octets:

1) Slot-time is a fundamental link parameter with multiple uses:

A) Slot-time is used by each Link Master DLE connected to the link in determining the amount of time for which that DLE monitors the link for inactivity before sending a CLAIM LAS (CL) DLPDU. Slot-time is defined such that the nominal link-inactivity monitoring periods of two DLEs which:

— have consecutive NODE DL-addresses; and

— which do not hold any token

differ by exactly one slot-time.

B) Slot-time is used by each Basic, Link Master, or Bridge DLE on the local link to compute the durations of all other periods of link inactivity which the DLE must monitor. All such durations are specified as standardized or configured multiples of one slot-time.

2) Slot-time is a configured minimum upper bound on the maximum two-way asynchronism in immediate communications among interacting DLEs on the local link when trying to (re)initialize the link, maximized across all pairs of DLEs on that local link. Given this viewpoint, slot-time is an aggregate measure of the worst-case implementation delays within the intervening media, the PhL, and the PhL/DLL interfaces, all of which limit the rapidity of two-way DLE interaction on the local link.

Slot-time shall be computed as the sum of:

a) the worst-case two way propagation delays through the intervening media and intervening PhEs, such as repeaters, between any two PhEs associated with their respective DLEs on the local link, measured between the point of connection to the local link of each of those PhEs, including worst-case internal logic and analog circuit delays in any intermediate repeaters;

b) the maximum-inactivity-to-claim-LAS-delay as defined in 3.3.25.1; and

c) a safety factor, which is used to account for:

1) the difference in two relevant internal delays consisting of

i) the delay between

A) the presentation of the first non-silent PhPDU to the DLE's associated PhE at that PhE's point of attachment to

B) the indication of start of activity from that PhE to that DLE; and

ii) the delay between

A) the presentation of the last non-silent PhPDU to the DLE's associated PhE at that PhE's point of attachment to

B) the indication of end of activity from that PhE to that DLE;

2) the rate differences in the internal timer clocks among the DLEs on the local link;

3) the limited resolution of the measurements and potential measurement errors; and

4) any extra delay needed to make the sum of (a),(b) and (c) round to an integral multiple of the transmission period of one octet.

NOTES —

1. These delays are defined in a manner that permits their measurement.

2. Finer resolution than one octet is not possible without knowledge of the specific associated PhL, because Ph-timing comes from the PhL and the formal PhL-DLL interface provides only octet timing see 5.4).

3. The definitions of 3.3.25 through 3.3.28 are not recursive, because 3.3.25 is based on a different measurement than 3.3.26 through 3.3.28. The definitions of 3.3.26 through 3.3.28 are in units of one slot-time to reduce the complexities of implementations of this protocol. As a result, a single 8-bit hardware timer, prescaled by the slot-time, enabled on bus inactivity, with usage-specific stopping and retriggering conditions, can provide the required functionality for all of the slot-time-based timers of this protocol.

3.3.25.1 maximum-inactivity-to-claim-LAS-delay A configuration parameter of each local link, its minimum value is the worst-case internal delays of any Link Master DLE and associated PhEs connected to that link. This internal delay is the sum of the following two components:

a) the internal delay between

1) the moment of presentation of the last non-silent PhPDU of a transmission to the DLE's associated PhE at that PhE's point of connection to the local link; and

2) the resulting start of the DLE's internal timer which monitors link inactivity; and

b) the internal delay between

1) the expiration of that timer; and

2) the resulting moment of presentation of the first non-silent PhPDU of a transmission of the required CLAIM LAS (CL) DLPDU by the DLE's associated PhE at that PhE's point of connection to the local link.

The unit of measurement of this aggregate delay is one-eighth of the transmission period of one octet (that is, one nominal "bit-period"). The range of values of this parameter is 1 to 4095 nominal "bit-periods."

3.3.26 maximum-response-delay: A measure, in units of one slot-time, greater than the worst-case period of **local** PhE-inactivity which can be observed by a DLE

a) which has just received a delegated token, before initiating transmission of a DLPDU; and

b) which has just received a reply token, before initiating transmission of the required immediate reply DLPDU.

A DLE's minimum required value for maximum-response-delay shall be determined by measuring at the responding DLE the delay between concluding the reception of a requesting DLPDU and initiating the transmission of the immediately-subsequent responding DLPDU.

Maximum-response-delay is a configuration parameter of each local link, and shall have a value of 1 to 11. When multiplied by the duration of one slot-time, the product shall be at least as large as the largest of the maximum-response-delay values required by each of the current or anticipated DLEs on the local link.

3.3.27 immediate-response-recovery-delay: A measure, similar to maximum-response-delay, of the worst-case period of **peer** PhE-inactivity which can be observed

a) by a DLE holding a delegated token while awaiting an immediate response from another correctly-functioning DLE on the local link; and

b) by any other DLE on the local link, after receiving the initiating transaction of a two-DLPDU transaction, before detecting the link activity caused by transmission of that immediate response DLPDU; and

c) by the LAS DLE while waiting for the initial link activity after

— a reply token (via a PROBE NODE-ADDRESS (PN) DLPDU;

— a delegated token (via a PASS TOKEN (PT) or EXECUTE SEQUENCE (ES) DLPDU); and

— the scheduler token (via a TRANSFER LAS (TL) DLPDU)

to another correctly-functioning DLE on the local link.

More formally, immediate-response-recovery-delay provides a bound for the worst-case delay between receipt of the Ph-DATA confirm primitive for a Ph-DATA request primitive whose PhIDU specified END-OF-DATA-AND-ACTIVITY (see 5.4.3), and subsequent receipt of the next START-OF-ACTIVITY PhIDU (see 5.4.3), that can be observed by the sending DLE on the local link during

1) an initiator-responder interaction, in which a transaction-initiating DLE sends a DLPDU requiring an immediate reply to a responding DLE, and that responding DLE replies by sending a second DLPDU; and

2) a token-passing interaction, in which the LAS DLE sends a delegated token DLPDU to the next intended token-holding DLE, or the scheduler token to a successor (as LAS) DLE, and that addressed DLE responds by sending a second DLPDU.

The value of immediate-response-recovery-delay, in units of one slot-time, shall be (maximum-response-delay + 1).

3.3.28 token-recovery-delay: A measure, similar to maximum-response-delay, greater than the worst-case period of **peer** PhE-inactivity which can be observed by the LAS DLE while another correctly-functioning DLE is using a token.

More formally, token-recovery-delay provides a bound greater than the worst-case delay between receipt of an END-OF-ACTIVITY or END-OF-DATA-AND-ACTIVITY PhIDU (see 5.4.4) and subsequent receipt of the next START-OF-ACTIVITY PhIDU, that can be observed by any DLE on the local link during normal link operation. This delay can be exceeded only when a DLE fails while holding a delegated or scheduler token.

The value of token-recovery-delay, in units of one slot-time, may be any value between (maximum-response-delay + 3) and 14.

3.3.29 link-id: The two-octet primary identifier for the local link, within the extended link, whose values are constrained as specified in Annex A/A.2.1.

3.3.30 node-id: The one-octet primary identifier for the DLE on the local link, whose values are constrained as specified in Annex A/A.2.2. A value of zero is also permitted, which inhibits all transmission by this DLE.

3.3.31 delocalization: The DLE-internal process of converting synonymous DL-addresses to a canonical form for transmission or during reception (see 6.2.2).

3.3.32 fractional-duty-cycle (FDC): A DLE which is not continuously attentive to received signaling, usually found in an end-system with an austere electrical power budget. Such a DLE generally requires the assistance of a bridge (DL-relay) DLE to communicate with other DLEs on the same link, to provide for cases where the two DLEs are not concurrently active.

Three classes of FDC DLE have been identified:

class A — FDC DLEs which are totally inattentive to their local links during their “sleep” periods, and control the timing of their temporary “re-awakening” to link communications solely on the basis of internal DL-time or some other measurement;

class B — FDC DLEs which monitor the link during their periods of “sleep” for a particular wakeup DLPDU (see 7.21) addressed specifically to the DLE’s NODE DL-address, “re-awakening” upon receipt of such a DLPDU; and

class C — FDC DLEs which monitor the link during their periods of “sleep” for any DLPDUs addressed specifically to the DLE’s NODE DL-address, “re-awakening” upon receipt of such a DLPDU.

NOTE — DLEs which “sleep” between DLPDUs, but that respond to all DLPDUs addressed to any of their DL-addresses, are not FDC DLEs.

3.3.33 DL-address user-request queue: A multi-priority FIFO-within-priority queue, associated with a specific DL-address of a DLE, of in-process DLS-user-requests. This queue is partitioned into three disjoint sections:

1) those requests which have already been transmitted but remain available for retransmission, confirmation, or both — that is, which have not reached their maximum confirm delay and which

— are still within the DLCEP’s transmit window; or

— are awaiting an acknowledging response DLPDU at the DLSAP;

2) those requests which are ready for transmission but have not been completely transmitted; and

3) those requests which are either awaiting a DL-COMPEL-SERVICE request to release them for transmission, or are outside the DLCEP’s transmit window, or both.

NOTES —

1. Section 1 of the queue contains DLS-user requests which may have been successfully communicated to another DLS-user. Retransmission of all or part of the DLS-user-data associated with these requests may be required, and can be attempted before the requests are purged from this partition (by reset or peer acknowledgment on a peer DLC; by reset or user-request timeout or the need for sequence-number reuse on a multi-peer DLC).

2. Section 2 of the queue contains DLS-user requests which are ready for transmission, but could not have been completely communicated to another DLS-user.

3. Section 3 of the queue can be non-empty only when the DL-scheduling-policy for the DL-address is EXPLICIT, or when the number of queued DLSDUs exceeds the DLCEP’s transmit window, or both.

4. Members of a given priority are advanced from one section to another until they are removed from the queue. (In practice, the section partitions may be moved, rather than the members.) The FIFO ordering within each priority is strictly maintained.

One such queue is associated with each DLSAP-address, each peer or publisher DLCEP-address, and each subscriber's DLCEP (which can be considered to be associated with an implicit DLCEP-address) of the DLE, and with the DLE's NODE DL-address (see 3.3.17and 6.2.2.3).

3.3.34 DLE unscheduled-service queue: A multi-priority FIFO-within-priority queue of

a) **references** to DL-address user-request queues (see 3.3.33);

NOTE – See 5.7.1.17(a) for a more elaborate definition.

b) **references** to locally-scheduled active sequences (see Annex B) resulting from DL-SCHEDULE-SEQUENCE requests (see 9.4.3.1); and

c) DT DLPDUs containing DLS-user-data which are delayed responses to received CD and ED DLPDUs, queued in support of the DL-UNITDATA-EXCHANGE service.

NOTE — Since this is a multi-priority FIFO-within-priority queue, members are removed in priority order, and within a single priority, in FIFO order.

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

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

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

NOTE — These concepts also support migration from previous national standards.

4 Symbols and abbreviations

4.1 Data units

BPDU	Bridge-protocol-data-unit, used in the ISO/IEC 10038 inter-bridge protocol
DLPDU	DL-protocol-data-unit
DLSDU	DL-service-data-unit
pDLSDU	partial DL-service-data-unit — one segment of a multi-segment DLSDU
PhIDU	Ph-interface-data-unit
SPDU	Support-protocol-data-unit, used to support the full DL-protocol

4.2 Local variables, timers, counters and queues

V(ST)	slot-time	See 5.7.1.1
V(PhLO)	per-DLPDU-PhL-overhead	See 5.7.1.2
V(MRD)	maximum-response-delay	See 5.7.1.3

V(IRRD)	immediate-response-recovery-delay	See 5.7.1.4
V(MRC)	maximum-retry-count	See 5.7.1.5
V(NRC)	network-retry-count	See 5.7.1.6
V(NDL)	network-DLPDU-lifetime	See 5.7.1.7
V(TN)	this-node	See 5.7.1.8
V(TL)	this-link	See 5.7.1.9
V(MEP)	DL-MAC-address-embedding-prefix	See 5.7.1.10
C(RD)	remaining-duration down-counter	See 5.7.1.11
V(MID)	minimum-inter-DLPDU-delay	See 5.7.1.12
T(IRRD)	immediate-response-recovery-delay monitor	See 5.7.1.13
V(RA)	reply-address	See 5.7.1.14
V(OTA)	outstanding-transaction-array	See 5.7.1.15
V(LTI)	last-transaction-index	See 5.7.1.16
Q(US)	unscheduled-service queue	See 5.7.1.17
V(RID)	random-identifier	See 5.7.1.18
C(NT)	node-time up-counter	See 5.7.1.19
V(LSTO)	local-link-scheduling-time-offset	See 5.7.1.20
V(DLTO)	DL-time-offset	See 5.7.1.21
V(TQ)	time-quality	See 5.7.1.22
V(MD)	measured-delay	See 5.7.1.23
V(LN)	LAS-node	See 5.7.1.24
V(TSC)	time-synchronization-class	See 5.7.1.25
T(TDP)	time-distribution-period monitor	See 5.7.1.26
V(TSL)	time-source-link	See 5.7.1.27
P_U(SDUL)	DLSDU-length request parameter	See 5.7.2.1
P_U(SDU)	DLSDU request parameter	See 5.7.2.2
P_U(MCD)	maximum-confirm-delay parameter	See 5.7.2.3
T_U(MCD)	maximum-confirm-delay monitor	See 5.7.2.4
Q_A(UR)	user-request queue	See 5.7.3.1
V_C(ST)	DLCEP state	See 5.7.4.1
V_C(NP)	negotiated DLCEP parameters	See 5.7.4.2
V_C(N)	next sequence number to assign to a DLSDU	See 5.7.4.3
V_C(R)	maximum non-transmittable DLSDU sequence number	See 5.7.4.4
V_C(A)	maximum acknowledged DLSDU sequence number	See 5.7.4.5

V_C(M)	minimum untransmitted DLSDU sequence number	See 5.7.4.6
V_C(MS)	minimum untransmitted segment number	See 5.7.4.7
$V_{C,K}$(SS)	to-be-sent segments of a DLSDU	See 5.7.4.8
$T_{C,K}$(SS)	sent-segments monitor for a DLSDU	See 5.7.4.9
V_C(L)	last-reported DLSDU sequence number	See 5.7.4.10
V_C(H)	highest-detected DLSDU sequence number	See 5.7.4.11
V_C(HS)	highest-detected segment number of the highest-detected DLSDU sequence number	See 5.7.4.12
$V_{C,K}$(MRS)	missing received segments of a DLSDU	See 5.7.4.13
$V_{C,K}$(RRS)	retransmission-request required segments of a DLSDU	See 5.7.4.14
$T_{C,K}$(RRS)	retransmission request monitor for a DLSDU	See 5.7.4.15
T_C(RAS)	residual activity stimulus	See 5.7.4.16
T_C(RAM)	residual activity monitor	See 5.7.4.17
V_C(TNA)	DL-time of last network access	See 5.7.4.18
V_B(TW)	DL-time of last buffer write	See 5.7.4.19
V_B(TP)	DL-time of production	See 5.7.4.20
V_B(TS)	Timeliness-status of buffer write	See 5.7.4.21
V(DTA)	delegation-address	See 5.7.5.1
V(LL)	local-link live-list	See 5.7.5.2
V(TCL)	token-circulation list	See 5.7.5.3
V(ENRL)	expected-non-response list	See 5.7.5.4
V(MST)	maximum-scheduled-traffic	See 5.7.5.5
V(MSO)	maximum-scheduling-overhead	See 5.7.5.6
V(DMDT)	default-minimum-token-delegation-time	See 5.7.5.7
V(DTHT)	default-token-holding-time	See 5.7.5.8
V(LTHT)	link-maintenance-token-holding-time	See 5.7.5.9
V(MTHA)	maximum-token-holding-time-array	See 5.7.5.10
V(TTRT)	target-token-rotation-time	See 5.7.5.11
V(ATRT)	actual-token-rotation-time	See 5.7.5.12
V(RTHA)	remaining-token-holding-time-array	See 5.7.5.13
V(NTHN)	next-token-holding-node	See 5.7.5.14
V(FUN)	first unpolled node id	See 5.7.5.15
V(NUN)	number of consecutive unpolled node ids	See 5.7.5.16
P(TRD)	token-recovery-delay	See 5.7.5.17

V(TDP)	time-distribution-period	See 5.7.5.18
V(MICD)	maximum-inactivity-to-claim-LAS-delay	See 5.7.5.19
V(LDDP)	LAS-data-base-distribution-period	See 5.7.5.20
V(ML)	maximum-link	See 5.7.6.1

4.3 DLPDU classes

CA	Compel Acknowledgment	See 7.4
CD	Compel Data	See 7.5
CL	Claim LAS	See 7.19
CT	Compel Time	See 7.9
DC	Disconnect Connection	See 7.2
DT	Data	See 7.7
EC	Establish Connection	See 7.1
ED	Exchange Data	See 7.6
ES	Execute Sequence	See 7.16
IDLE	Idle	See 7.22
PN	Probe Node-address	See 7.13
PR	Probe Return	See 7.14
PT	Pass Token	See 7.15
RC	Reset Connection	See 7.3
RI	Request Interval	See 7.18
RQ	Round-trip-delay Query	See 7.11
RR	Round-trip-delay Reply	See 7.12
SR	Status Response	See 7.8
TD	Time Distribution	See 7.10
TL	Transfer LAS	See 7.20
RT	Return Token	See 7.17
WK	Wakeup	See 7.21

4.4 Miscellaneous

DL-	Data Link layer (as a prefix)
DLC	DL-connection
DLCEP	DL-connection-end-point
DLE	DL-entity (the local active instance of the Data Link layer)
DLL	DL-layer

DLM-	DL-management (as a prefix)
DLMS	DL-management-service
DLP-	DL-protocol (as a prefix)
DLPCI	Data Link-protocol-control-information
DLPDU	DL-protocol-data-unit
DLS	DL-service
DLSAP	DL-service-access-point
DLSDU	DL-service-data-unit
DLSEP	DL-schedule-end-point
FC	Frame Control (first octet of each DLPDU)
FCS	Frame Check Sequence
FDC	Fractional Duty Cycle (type of DLE)
FIFO	First-in first-out (queuing method)
LAS	Link Active Scheduler
LLC	Logical Link Control
LM	Link Master
MAC	Medium Access Control
OSI	Open Systems Interconnection
Ph-	Physical layer (as a prefix)
PhE	Ph-entity (the local active instance of the Physical layer)
PhICI	Physical-interface-control-information
PhID	Physical-interface-data
PhL	Ph-layer
PhS	Ph-service
PhSAP	Ph-service-access-point
QoS	Quality of Service
SLAE	Systems-load-application-entity
SMAE	Systems-management-application-entity

5 Overview of the DL-protocol

5.1 Three-level model of the DLL

The DLL is modeled as

a) a low-level of path-access-and-scheduling functions, which supports DL-service functions.

b) an intermediate-level of bridge-operation functions, which in turn supports DL-service functions.

c) a higher-level of connection-mode and connectionless data transfer, bridge-coordination, and DL-service functions.

Interoperating with all three levels are DL-management functions, including bridge and redundant-path management functions where relevant.

NOTES —

1. The term “sublayer” is not appropriate for describing these levels, since ISO 7498 requires that when multiple sublayers are defined, all but one of them must be optional.

2. This three-level partitioning closely resembles the partitioning into lower-level “MAC”, intermediate-level “bridge-operation”, and higher-level “LLC” functions found in the ISO/IEC LAN standards [ISO/IEC TR 18802], with two significant differences:

A) This protocol’s low-level functionality is contained entirely within the Data Link layer as specified by ISO 7498. In contrast, the ‘MAC’ functionality of the ISO/IEC LAN protocols spans the lower part of the OSI Data Link layer and the upper part of the OSI Physical layer.

B) This protocol employs a single level of DL-addressing within the Data Link layer. In contrast, the ISO/IEC LAN protocols employ two levels of DL-addressing, one within the “MAC” and “bridge-operation” functionality and a second within the “LLC” functionality.

5.1.1 Path access and scheduling level

a) The path-access-and-scheduling level forms each DLPDU from DL-protocol control information and DLS-user-data; computes and appends an appropriate frame check sequence (FCS); and passes the whole as a sequence of PhIDUs (see 5.4) to the PhE for transmission to peer PhEs for reporting to peer DLEs.

In some cases it also appends the low-order three octets either

— of the current value of the local node-timer, C(NT); or

— of a computed value minus the current value of the local node-timer, C(NT),

during DLPDU formation, immediately preceding the appended FCS.

b) The path-access-and-scheduling level receives a sequence of PhIDUs from the PhE, concatenates those PhIDUs into a received DLPDU, computes a frame check sequence over the entire sequence of received data, and checks for the proper residual value. The first octet of the received sequence is examined to determine the type of the DLPDU, and an attempt is made to parse that DLPDU into its DL-protocol control information and DLS-user-data components. If the FCS residual was correct and the parse was successful, then the appropriate low-level actions are performed, possibly including reporting the parsed DLPDU to a higher level.

In some cases the value of the low-order three octets of the local node-timer, C(NT), at the time of receipt, is appended to this parsed DLPDU.

c) The path-access-and-scheduling level provides the basic functions of both responder and initiator. As a responder, it provides the sequencing functions necessary

1) for receiving a DLPDU, possibly conveying a reply token; and

2) in that latter case (of the received reply token), for sending a DLPDU as an immediate reply to the just-received DLPDU.

As an initiator, it provides the sequencing functions necessary for

3) receiving a delegated token;

4) sending one or more DLPDUs, including those requiring an immediate reply;

5) receiving such an immediate reply, or inferring its absence; and

6) returning a delegated token.

d) The path-access-and-scheduling level provides the low-level scheduling functionality required for scheduling DLPDU transmissions on a specific path, including any interactions with the local link's link active scheduler (LAS) to coordinate the schedule with other DLEs or to request needed path transmission capacity.

The actions of (a) and (b) are augmented within a bridge (relay DLE) to permit the retransmission of a received sequence of data octets, including the received FCS, with possible constrained alterations of the first octet and compensating alterations (see 6.1.1.3) of the received FCS, prior to retransmission.

The actions of (c) and (d) are based in part on two request-management and scheduling queues:

i) a DL-address-specific request queue (see 3.3.33), associated with the sending address, which is used to manage DLS-user requests originated from that DL-address; and

ii) the common DLE unscheduled-service queue (see 5.7.1.17), which is used to manage the servicing of unscheduled requests upon receipt of the "circulated" token.

Some of the more complex scheduling functionality of this level requires, and uses, the services of the upper level (see 5.1.3) by acting as a DLE-internal quasi-DLS-user.

5.1.2 Bridge-operation level

The bridge-operation level provides the intermediate-level functionality of

a) logically inter-connecting multiple local links into a single extended-link by physically interconnecting multiple paths;

b) serving as a (possibly-distributed) "time-space-time" switch:

1) providing DLPDU store-and-forward functions to permit communication between DLEs on the extended link which could not otherwise communicate; and

NOTE — This includes the coordination with fractional-duty-cycle (FDC) DLEs (see 3.3.32) necessary to permit alternating periods of "FDC-DLE-awake" and "FDC-DLE-asleep" operation.

2) providing surrogate functions to permit delayed-reply interactions between DLEs on the extended link as an extrapolation of "immediate-reply" transactions on a local link;

c) providing a shared sense of DL-time throughout the extended link; and

d) coordinating local-link scheduling among two or more local links to provide any necessary multi-link coordination of scheduling within the extended link.

Much of the bridge-operation level of functionality is active only in "bridge" DLEs (see 5.6).

5.1.3 Connection-mode and connectionless data transfer, bridge-coordination, and DL-service level

The connection-mode and connectionless data transfer, bridge-coordination, and DL-service level provides the higher-level functionality of

a) managing all DLE interactions with the DLS-user, converting all DLS-user request and response primitives into the necessary sequence of DLE operations, and generating DLS-user indication and confirm primitives where appropriate;

b) managing the sequencing of each active DLCEP, including

1) determination of the type and sequence of DLPDUs to be transmitted from the DLCEP;

2) QoS negotiation;

3) determination of the DLPCI to be included in each DLPDU; and

4) segmentation and reassembly of large DLSDUs;

c) managing the sequencing of unitdata transactions which require a response from a peer DLE, including

1) determination of the DLPCI to be included in each DLPDU; and

2) correlation of a non-immediate reply DLPDU with the requesting transaction;

d) processing all inter-DLE state-information-distribution DLPDUs, such as TIME DISTRIBUTION (TD) (see 7.10 and 9.4.1) and bridge configuration (see Annex C) DLPDUs and LAS-backup SPDUs (see Annex B); and

e) managing all query-DLPDU/reply-DLPDU interactions with other DLEs, other than those which occur as an "immediate reply" and which are made on a reactive basis by the lower-level functions of 5.1.1.

These query-DLPDU/reply-DLPDU interactions also include computation of round-trip-delays (see 7.11, 7.12, and 9.4.1), support of remotely-initiated DL-SUBSCRIBER-QUERY (see 9.2.3) and DL-LISTENER-QUERY (see 9.3.2) requests, and inter-bridge exchanges of bridge state information (see Annex C).

5.2 Service provided by the DLL

The DLL provides connectionless data transfer services for limited-size DLSDUs, connection-mode data transfer services for limited-size DLSDUs, an internally-synchronized time service, scheduling services to control the time allocation of the underlying shared PhL service, and a DL(SAP)-address, queue and buffer management service.

NOTE — Table 10 of Part 3 of this International Standard shows many of the relationships among DLC QoS attributes.

Some relevant QoS attributes are as follows:

5.2.1 QoS — DLCEP class

Each DLCEP establishment request specifies the class of the DLCEP. The three choices for DLCEP-class are

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

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

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

5.2.2 QoS — DLCEP data delivery features

Both members of a peer DLC, or the publishing DLS-user of a multi-peer DLC, specify the data delivery features of the DLC's DLCEP(s). The five choices for DLCEP data delivery features, and their effects, are

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

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

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

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

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

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

5.2.3 QoS — DLL priority

All DLCEP establishment requests and responses, all connectionless data transfer requests, and many DL-scheduling requests, specify an associated DLL priority used in scheduling DLL data transfer services. This DLL priority also determines the maximum amount of DLS-user-data that can be conveyed in a single DLPDU. The three DLL priorities with their corresponding ranges of conveyable DLS-user-data (per DLPDU) are, from highest priority to lowest priority:

a) **URGENT** — ≤ 64 DLS-user octets per DLPDU;

b) **NORMAL** — ≤ 128 DLS-user octets per DLPDU;

c) **TIME-AVAILABLE** — ≤ 256 DLS-user octets per DLPDU.

NOTES —

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

2. DLC establishment and DLC release DLPDUs which are sent at TIME-AVAILABLE priority are restricted to convey no more DLS-user-data than would be permitted at NORMAL priority — 128 octets.

5.2.4 QoS — DLPDU authentication

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

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

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

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

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

NOTE 2 — These last two aspects are of particular importance in safety systems.

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

1) **ORDINARY** — each DLPDU shall include the minimum amount of addressing information necessary;

2) **SOURCE —** each DLPDU shall include a source DL-address where possible; and

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

5.2.5 QoS — DLL maximum confirm delay

Each DLCEP establishment request, and each response, specifies upper bounds on the maximum time duration permitted for the completion

a) of a DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY primitives, and, separately;

b) of a DL-DATA primitive.

Each connectionless service request specifies an upper bound on the maximum time duration permitted for the completion

c) of a locally-confirmed DL-UNITDATA primitive, and, separately;

d) of a remotely-confirmed DL-UNITDATA primitive, a DL-LISTENER-QUERY primitive, or an instance of the DL-UNITDATA-EXCHANGE service.

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

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

The intervals specified are the maximum permissible delays

1) between the issuing of the specified request primitives and the issuing of the corresponding confirm primitives; and

2) between the initiation and completion of a single instance of the specified publishing or unitdata-exchange service.

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

Failure to complete a DL-CONNECT or DL-RESET request within the specified interval shall result in a DLS-provider initiated release of the DLCEP, and possibly of the DLC.

5.2.6 QoS — DL-scheduling-policy

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

The two choices are:

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

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

NOTE — Scheduling of DLPDU transmission to support the DL-CONNECT, DL-RESET and DL-DISCONNECT services, and to support responder-deferred replies in the DL-UNITDATA-EXCHANGE and remotely-confirmed DL-UNITDATA services, is always IMPLICIT. Scheduling of DLPDU transmission to initiate the DL-UNITDATA-EXCHANGE service is always EXPLICIT.

5.2.7 QoS — maximum DLSDU sizes

Each DLC/DLCEP establishment request, and each response, specifies an upper bound on the size (in octets) of DLSDUs which will be offered for transmission, and an upper bound on the size of DLSDUs which are acceptable for reception.

For peer DLCs, the negotiated maximum DLSDU size shall be determined independently for each direction of data transfer as the smallest of that offered by the sender, that permitted by the sender's local DL-management, that permitted by the receiver's local DL-management, and that permitted by the receiver.

The sender's offered size may be any value between zero and 16 times the maximum number of DLS-user octets per DLPDU, as specified in 5.2.3. The receiving DLE and all DL-management agents shall choose their maximum permitted sizes from the following list of sizes: 0, 64, 128, $256 \times N$ where $1 \le N \le 16$.

NOTES —

1. The maximum size DLSDU supported by this protocol is 16 times the maximum number of octets of DLS-user-data conveyable in a single DLPDU, as determined by the DLC's DLL Priority (see 5.2.3).

2. The set of maximum permitted DLSDU sizes is limited to the above small list of values to promote interoperation. The sender's maximum specified size is permitted to take any value within the range permitted by the DLC's DLL priority to facilitate optimization of the shared communications capacity of the DLL.

3. A value of zero (0) corresponds to the choice of simplex service, as specified by the DLS-user by the choice NONE as described in 5.2.2(e).

For multi-peer DLCs, the negotiated maximum DLSDU size shall be the smaller of that offered by the publisher and that permitted by the publisher's local DL-management. For subscribers of multi-peer DLCs, the DLC shall be refused by the DLS-provider (the subscriber's DLE) if the maximum DLSDU size established by the publisher is larger than the smaller of that permitted by the subscriber and that permitted by the subscriber's local DL-management.

The publisher's offered size in the publisher-to-subscribers direction may be any value between zero and 16 times the maximum number of DLS-user octets per DLPDU, as specified in 5.2.3. The publisher's offered size in the subscribers-to-publisher direction may be any value between zero and the maximum number of DLS-user octets per DLPDU, as specified in 5.2.3.

The subscribers and all DL-management agents shall choose their maximum permitted sizes from the following list of sizes: 0, 64, 128, $256 \times N$ where $1 \le N \le 16$.

NOTES —

4. The maximum size DLSDU supported by this protocol in the publisher-to-subscribers direction is 16 times the maximum number of octets of DLS-user-data conveyable in a single DLPDU, as determined by the DLC's DLL Priority (see 5.2.3).

5. The maximum size DLSDU supported by this protocol in the subscribers-to-publisher direction is the maximum number of octets of DLS-user-data conveyable in a single DLPDU, as determined by the DLC's DLL Priority (see 5.2.3).

6. The set of maximum permitted DLSDU sizes is limited to the above small list of values to promote interoperation. The publisher's maximum specified DLSDU size is permitted to take any value within the range permitted by the DLC's DLL priority to facilitate optimization of the shared communications capacity of the DLL.

7. A value of zero (0) corresponds to the choice of simplex service, as specified by the DLS-user by the choice NONE as described in 5.2.2(e).

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

5.2.8 QoS — DLCEP and DLSAP-address buffer-and-queue bindings

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

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

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

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

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

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

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

Each DLSAP-address bind request can bind up to six local retentive buffers or non-retentive buffers or specified-depth FIFO queues, created by DL-CREATE buffer and queue management primitives (or by DL-management), to the DLSAP-address:

d) One buffer or queue can be bound to the sending direction of a DLSAP-address at each priority to convey DLSDUs from the DLS-user to the DLS-provider. Buffers can be bound only to DLSAP-addresses whose DLSAP-role is INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER. Queues can be bound only to DLSAP-addresses whose DLSAP-role is BASIC.

e) One buffer or queue can be bound to the receiving direction of a DLSAP-address at each priority to convey DLSDUs from the DLS-provider to the DLS-user. Buffers can be bound only to DLSAP-addresses whose DLSAP-role is INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER. Queues can be bound to all DLSAP-addresses.

5.2.8.1 Binding to a buffer

When a sending buffer is bound to a DLCEP, or to a DLSAP-address and DLL priority, by a DLS-user,

a) a DL-PUT request primitive overwrites the buffer with a DLSDU, or may set the buffer empty.

NOTE — After creation, the buffer is empty.

b) a DL-COMPEL request primitive, specifying either

1) a DLCEP; or

2) unitdata-exchange at a DLSAP-address

causes the transmission, at the first opportunity, of the then most recent contents of the buffer; the primitive does not itself specify a DLSDU.

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

d) a DL-UNITDATA-EXCHANGE indication primitive notifies the single DLS-user attached to the specific DLSAP-address from which the buffer was transmitted (and to which the buffer is bound) that the buffer was just transmitted on that DLSAP-address, and emptied if it is non-retentive (BUFFER-NR).

When a receiving buffer is bound to a DLCEP, or to a DLSAP-address and DLL priority, by a DLS-user,

e) a DL-BUFFER-RECEIVED or DL-UNITDATA-EXCHANGE indication primitive notifies the DLS-user of the overwriting of the buffer by a newly-received DLSDU; the primitive does not itself specify a DLSDU.

f) a DL-GET request primitive copies the DLSDU from the buffer, and empties the buffer if it is non-retentive (BUFFER-NR).

Multiple concurrent output bindings to a retentive buffer are permitted as an implementation and conformance (see 11.1) option, but are not required by this protocol.

5.2.8.2 Binding to a FIFO queue

When a sending FIFO queue of maximum depth *K* is bound to a DLCEP, or to a DLSAP-address and DLL priority, by a DLS-user,

a) a DL-PUT request primitive is not permitted.

b) a DL-DATA or DL-UNITDATA request primitive attempts to append a DLSDU to the queue, but fails if the queue already contains *K* DLSDUs. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.

When a receiving FIFO queue of maximum depth *K* is bound to a DLCEP, or to a DLSAP-address and DLL priority, by a DLS-user,

c) a DL-GET request primitive attempts to remove a DLSDU from the queue, but fails if the queue is empty.

d) a DL-DATA or DL-UNITDATA or DL-UNITDATA-EXCHANGE indication primitive notifies the receiving DLS-user of the result of appending a newly-received DLSDU to the receive queue; the primitive does not itself specify a DLSDU.

Multiple concurrent input bindings to a FIFO queue are permitted as an implementation option, but are not required by this protocol.

5.2.8.3 Default bindings

When these binding options are not specified, or for DLS-primitives for which explicit binding is not applicable, the conventional implicitly-queued sending and direct receiving interfaces between DLS-user and DLS-provider are employed. In this case each DL-DATA and DL-UNITDATA primitive conveys a DLSDU, and each DL-CONNECT, DL-DISCONNECT and DL-RESET primitive may convey a DLSDU:

a) DL-PUT and DL-GET request primitives are not permitted.

b) A DL-DATA or DL-UNITDATA or DL-DISCONNECT request primitive, or a DL-CONNECT or DL-RESET request or response primitive, issued by the sending DLS-user attempts to append a DLSDU to the implicit queue, but fails if the queue is full. If the append operation was successful, then the DLSDU will be transmitted at the first opportunity, after all preceding DLSDUs in the queue.

c) A DL-DATA or DL-UNITDATA or DL-DISCONNECT indication primitive, or a DL-CONNECT or DL-RESET indication or confirm primitive, notifies a receiving DLS-user of a newly-received DLSDU, and conveys that DLSDU to the DLS-user. No apparent queuing is provided within the DLL.

5.2.9 DL-timeliness

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

Each DLCEP establishment request, and each response, can specify DL-timeliness criteria which are to apply to information sent from, or received into, buffers at that DLCEP. Four types of DL-timeliness can be supported: RESIDENCE timeliness, UPDATE timeliness, SYNCHRONIZED timeliness, and TRANSPARENT timeliness. All four types of timeliness, and the case where there is no timeliness, are shown in Figure 4 of Part 3 of this International Standard:

a) **RESIDENCE** timeliness is an assessment based upon the length of time that a DLS-user datum has been resident in a buffer, which is the time interval between

1) the moment when the buffer is written (by a DL-PUT request primitive, or by reception into the buffer at a DLCEP); and

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

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

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

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

1) the moment of occurrence of a multi-DLE synchronizing event (a DL-BUFFER-RECEIVED indication or DL-BUFFER-SENT indication); and

2) the moment when the buffer is written (by a DL-PUT request primitive, or by reception into the buffer at a DLCEP);

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

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

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

1) the moment of occurrence of a multi-DLE synchronizing event (a DL-BUFFER-RECEIVED indication or DL-BUFFER-SENT indication);

2) the moment when the buffer is written (by a DL-PUT request primitive, or by reception into the buffer at a DLCEP); and

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

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

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

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

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

Where a buffer read or write occurs over a significant time interval, and not just as a momentary event, then the overall timeliness of the read or write operation shall be computed as the timeliness at the beginning of the read or write operation, logically ANDed with the timeliness at the end of the read or write operation, all using the same timeliness criteria.

5.2.10 Remote-DLE-confirmed

Each unitdata transfer request can specify whether confirmation of receipt of the associated DLSDU by the (implicitly addressed) remote DLE is required. Its permissible values are **TRUE** and **FALSE**.

NOTE — Selection of the value TRUE inevitably uses more link capacity than does selection of the value FALSE.

5.3 DL-addresses

DL-addresses are used as DLSAP-addresses and group DL-addresses, as DLCEP-addresses, and as DLSEP-addresses. DL-addresses conform to the structure and coding limitations specified in Annex A.

NOTE — DL-addresses are also addressed to some extent in 6.2.

5.4 Service assumed from the Physical Layer

This sub-clause defines the assumed Physical Service (PhS) primitives and their constraints on use by the DLE.

NOTE — Proper layering requires that an (N+1)-layer entity not be concerned with, and that an (N)-service interface not overly constrain, the means by which an (N)-layer provides its (N)-services. Thus the Ph-service interface does not require DLEs to be aware of internal details of the PhE (for example, preamble, postamble and frame delimiter signal patterns, number of bits per baud), and should not prevent the PhE from using appropriate evolving technologies.

5.4.1 Assumed primitives of the PhS

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

5.4.1.1 PhS characteristics reporting service

The PhS is assumed to provide the following service primitive to report essential PhS characteristics used in DLL transmission, reception, and scheduling activities:

Ph-CHARACTERISTICS indication (minimum-data-rate, framing-overhead)

where

minimum-data-rate — specifies the effective minimum rate of data conveyance in bits/second, including any timing tolerances, of any PhE on the local link.

NOTE 1 — A PhE with a nominal data rate of 1 Mbit/s ± 0,01% would specify a minimum data rate of 0,9999 Mbit/s.

framing-overhead — specifies the maximum number of bit periods, where period = , used in any transmission for PhPDUs which do not directly convey data (for example, PhPDUs conveying preamble, frame delimiters, postamble, inter-frame “silence”, and so on).

NOTE 2 — If the framing overhead is F and two DL message lengths are L_1 and L_2, then the time to send two immediately consecutive messages of lengths L_1 and L_2 will be at least as great as the time required to send one message of length $L_1 + F + L_2$.

If this framing-overhead is more than the DLE’s configured per-DLPDU-PhL-overhead, V(PhLO), then the DLE shall report this discrepancy to DL-management and shall not issue Ph-DATA requests while the discrepancy exists.

NOTE 3 — This restriction prohibits DLE transmission while this discrepancy exists. The DLE’s local station management may remedy this discrepancy by reconfiguring V(PhLO) to a greater value.

5.4.1.2 PhS transmission and reception services

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

Ph-DATA request (class, data);

Ph-DATA indication (class, data); and

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 — transmission of the PhPDUs which precede Ph-user data should commence;

DATA — the single-octet value of the associated data parameter should be transmitted as part of a continuous correctly-formed transmission; and

END-OF-DATA-AND-ACTIVITY — the PhPDUs which terminate Ph-user data should be transmitted after the last preceding octet of Ph-user data, culminating in the cessation of active transmission;

For a Ph-DATA indication, its possible values are:

START-OF-ACTIVITY — reception of an apparent transmission from one or more PhEs has commenced;

DATA — the associated data parameter was received as part of a continuous correctly-formed reception;

END-OF-DATA — the ongoing continuous correctly-formed reception of Ph-user data has concluded with correct reception of PhPDUs implying END-OF-DATA;

END-OF-ACTIVITY — the ongoing reception (of an apparent transmission from one or more PhEs) has concluded, with no further evidence of PhE transmission; and

END-OF-DATA-AND-ACTIVITY — simultaneous occurrence of END-OF-DATA and END-OF-ACTIVITY;

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 which was received successfully (Ph-DATA indication).

status — specifies either success or the locally-detected reason for inferring failure.

The Ph-DATA confirm primitive provides the critical physical timing feedback necessary to inhibit the DLE from starting a second transmission before the first is complete. The final Ph-DATA confirm of a transmission shall not be issued until the PhE has completed the current transmission.

5.4.2 Notification of PhS characteristics

The PhE has the responsibility for notifying the DLE of those characteristics of the PhS which are relevant to DLE operation. This notification is accomplished by the PhE by issuing a single Ph-CHARACTERISTICS indication primitive at each of the PhE's PhSAPs at PhE startup.

5.4.3 Transmission of Ph-user-data

The PhE determines the timing of all transmissions. When a DLE has a DLPDU to transmit, and the DL-protocol gives that DLE the right to transmit, then the DLE shall send the DLPDU, including a concatenated FCS, by making a well-formed sequence of Ph-DATA requests, consisting of a single request specifying START-OF-ACTIVITY; followed by three to 300 consecutive requests, inclusive, specifying DATA; and concluded by a single request specifying END-OF-DATA-AND-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. A second Ph-DATA request should not be issued until after the Ph-DATA confirm corresponding to the first request has been received from the PhE.

5.4.4 Reception of Ph-user-data

The PhE reports a received transmission with a well-formed sequence of Ph-DATA indications, which shall consist of either

a) a single indication specifying START-OF-ACTIVITY; followed by consecutive indications specifying DATA; followed by a single indication specifying END-OF-DATA; and concluded by a single indication specifying END-OF-ACTIVITY; or

b) a single indication specifying START-OF-ACTIVITY; followed by consecutive indications specifying DATA; followed by a single indication specifying END-OF-DATA-AND-ACTIVITY; or

c) a single indication specifying START-OF-ACTIVITY; optionally followed by one or more consecutive indications specifying DATA; and concluded by a single indication specifying END-OF-ACTIVITY.

NOTE — This last sequence is indicative of an incomplete or incorrect reception. Detection of an error in the sequence of received PhPDUs, or in the PhE's reception process, disables further Ph-DATA indications with a class parameter specifying DATA, END-OF-DATA, or END-OF-DATA-AND-ACTIVITY until after both the end of the current period of activity and the start of a subsequent period of activity have been reported by Ph-DATA indications specifying END-OF-ACTIVITY and START-OF-ACTIVITY, respectively.

In the first two cases, the DLE concatenates the received data and then attempts to parse it into a DLPDU followed by a concatenated FCS. In the last case the DLE discards all reported data and reports the event to DL-management.

5.5 Functions of the DLL

5.5.1 Overview of functions

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. When used in a normal OSI environment, the necessary functions of the DLL are those specified in ISO/IEC 8886. When used in a Time Critical OSI environment, the necessary functions are a superset of those specified in ISO/IEC 8886; the enhancements are primarily in

a) the availability of a confirm primitive and a service confirmation deadline for each connection-oriented and connectionless data-transfer DL-service;

b) the addition of connectionless receiving-DLE-acknowledgment and two-way data-exchange services to the basic OSI data-transfer DL-service;

c) the ability to defer and schedule data-transfer DL-services;

d) the efficient distribution of DLS-user-data from a publishing DLS-user to a set of subscribing DLS-users;

e) the efficient convergence of independent DLS-user-data from a set of subscribing DLS-users, delivered to a single publishing DLS-user;

f) the provision of a synchronized sense of internal time among the DLEs and available to the DLS-users of the extended link; and

g) the standardized availability of local DL(SAP)-address, buffer and queue management capabilities.

5.5.2 Connectionless data transfer functions

The purpose of the connectionless data transfer functions is to transport DLSDUs of limited size between one DLS-user and one or more other DLS-users on the same link, without the requirement for establishing or maintaining a DLC with each of those other DLS-users. This purpose is achieved by means of transmission of DLPDUs providing transfer of a limited amount of user data to one or many DLS-users, with limited protection against loss of the DLSDU, or misordering of successively-transmitted DLPDUs. Both DLSDU transfer and DLSDU exchange services are provided.

5.5.3 Connection-oriented functions

Connection-oriented functions provide for the establishment, use, resynchronization and abrupt termination of a connection between DLS-users on an extended link. The type of the connection may be selected to support user-data flow either

a) bidirectionally between two peer DLS-users;

b) unidirectionally from one peer DLS-users to another;

c) bidirectionally between one publishing DLS-user and the set of zero or more subscribing DLS-users;

d) unidirectionally from one publishing DLS-user to the set of zero or more subscribing DLS-users; or

e) unidirectionally from the set of zero or more subscribing DLS-users to the publishing DLS-user.

The features of the connection may be selected to support transfer of DLSDUs of a negotiated maximum size, with either

1) reliable in-sequence non-duplicated delivery with reset on DLSDU loss;

2) reduced delay, potentially out-of-sequence but non-duplicated delivery, and reset on DLSDU loss;

3) minimal delay with in-sequence non-duplicated delivery, but with potential DLSDU loss;

4) minimal delay, unsequenced delivery, potential duplication, potential but improbable DLSDU misordering, and no notification of DLSDU loss; or

5) no transfer in one specified direction.

Only cases (4) and (5) are permitted for the subscribers-to-publisher direction of a multi-peer DLC.

5.5.3.1 Connection establishment phase

The purpose of the connection establishment phase is to:

a) establish a DLC between two DLS-users,

NOTE — The establishment of a publishing DLC is best modeled as the concurrent independent pair-wise establishment of the DLC between the common publisher and each separate subscriber.

b) determine QoS attributes of the DLC; and

c) distinguish between DLCs.

5.5.3.2 Data transfer phase

The purpose of the data transfer phase is to:

a) transport DLSDUs between two DLS-users connected by a DLC. This purpose is achieved by transmission of DATA (DT) DLPDUs (see 6.1), which may involve segmenting of DLSDUs for conveyance in multiple DLPDUs and reassembly by the destination DLE; and

b) resynchronize the flow of DLSDUs between the DLS-users, and notify those DLS-users of information loss after an unrecovered error.

These concepts are defined in Part 3 of this International Standard.

5.5.3.3 Connection termination phase

The purpose of the connection termination phase is to terminate abruptly a connection between two or more DLS-users and convey the reason for the termination.

5.5.4 Time synchronization function

The purpose of the time synchronization function is to provide a shared approximately-synchronized internal time reference for all DLS-users, consisting of two components:

a) a component which increases monotonically with time, with a value of zero at the startup of the local end system; and

b) a second component which, when added to the first, causes the sum to be approximately equal to the corresponding sums of other correctly-functioning DLEs on the extended link.

5.6 Functional classes

In this part of this International Standard, a DLE's functional class determines its capabilities for autonomous DLL activity, and thus the minimum complexity of conforming implementations. Each class includes all lower-numbered classes. The three functional classes, in order of increasing complexity, are

a) Basic;

b) Link Master (LM); and

c) Bridge.

All functional classes support all DLS-user services and are completely interoperable. The DLPDUs associated with elements of procedure for each functional class are specified in Table 1.

Table 1 — Correlation of DLPDUs with Functional Classes

Functional Class					
Basic	**Link Master**	**Bridge**	**Code**	**DLPDU Name**	**Reference**
R1 S1	R1 S1	R1 S1	EC	Establish Connection	7.1
R1 S1	R1 S1	R1 S1	DC	Disconnect Connection	7.2
R1 S1	R1 S1	R1 S1	RC	Reset Connection	7.3
R S	R S	R S	CA	Compel Acknowledgment	7.4
R S	R S	R S	CD	Compel Data	7.5
R S	R S	R S	ED	Exchange Data	7.6
R S	R S	R S	DT	Data	7.7
R —	R S	R S	SR	Status Response	7.8
—s2	R s2	R s2	CT	Compel Time	7.9
r2 —	R2 S	R2 S	TD	Time Distribution	7.10
R s2	R s2	R s2	RQ	Round-trip-delay Query	7.11
r2 S	r2 S	r2 S	RR	Round-trip-delay Reply	7.12
R —	R S	R S	PN	Probe Node DL-address	7.13
—S	R3 S	R3 S	PR	Probe Response	7.14
R —	R S	R S	PT	Pass Token	7.15
R —	R S	R S	ES	Execute Sequence	7.16
—S	R3 S	R3 S	RT	Return Token	7.17
—S	R3 S	R3 S	RI	Request Interval	7.18
——	R S	R S	CL	Claim LAS	7.19
——	R S	R S	TL	Transfer LAS	7.20
r —	r S	—S	WK	Wakeup	7.21
—s4	—s4	—s4	IDLE	Idle	7.22
——	——	——	spare	Spare	7.23

LEGEND
- r The specified class permits the ability to receive and act upon the specified type of DLPDU.
- R The specified class always provides the ability to receive and act upon the specified type of DLPDU.
- s The specified class permits the ability to send the specified type of DLPDU.
- S The specified class always provides the ability to send the specified type of DLPDU.
- — The specified class does not permit the specified type of DLPDU.

NOTES
1. Mandated support requires only minimal elements of procedure.
2. All must be supported when the DLE has a time-synchronism class (see 11.3.2) other than NONE.
3. Must respond to DLSDU contents when acting as LAS.
4. No class is required to receive the IDLE DLPDU; its intended receiver is a specialized link monitor or analyzer, beyond the scope of this standard.

5.6.1 Basic class

The Basic class includes the basic protocol elements of procedure necessary to:

a) provide interoperability when responding to DLPDUs sent by a DLS-peer or a bridge DLE;

b) initiate, reset and terminate DLCs with a peer DLE, to support the orderly conveyance of DLSDUs;

c) send and receive connectionless and connection-oriented DLSDUs, and to reply to received DLSDUs where required;

d) request services from the LAS;

e) execute an uninterrupted sequence of link operations; and

f) optimize local use of the link.

This class is the minimum necessary for fieldbus interoperability.

5.6.2 Link Master class

The Link Master class also includes the protocol elements of procedure necessary to:

a) cooperate with similar DLEs in establishing and sharing mastership of the link; and

b) detect the absence of an LAS on the link and activate the LAS functions within its own node,

and when providing the functions of the LAS,

c) maintain an ordered access to the shared link communications resource, responding to requests from other DLEs for use of that shared resource; and

d) serve as the source of internal time for the other DLEs on the link.

This class is necessary for autonomous operation on the link. At least one DLE on the link shall operate in this class.

5.6.3 Bridge (DL-relay) class

The Bridge class also includes the protocol elements of procedure necessary to:

a) enable communications between DLEs on a single link which are themselves periodically incapable of communicating directly on the link (that is, fractional duty cycle (FDC) DLEs), and in some cases serving as a surrogate for a DLE on the local link;

b) interconnect two or more local links, by bridging them into an extended link, and in some cases serving as a surrogate for a remote DLE on one of the bridge's local links; and

c) provide a common sense of DL-internal time coordinated across the extended link.

This class is necessary when interconnecting two or more local links to form a multi-link extended link, or when one or more DLEs on the local link are fractional duty cycle (FDC) DLEs. When a multi-link extended link exists, the individual local links shall be interconnected only by DLEs which operate in this class.

5.7 Local parameters, variables, counters, timers and queues

This part of this International Standard uses DLS-user request parameters P(…) and local variables V(…) as a means of clarifying the effect of certain actions and the conditions under which those actions are valid, local timers T(…) as a means of monitoring actions of the distributed DLS-provider and of ensuring a local DLE response to the absence of those actions, and local counters C(…) for performing rate measurement functions. It also uses local queues Q(…) as a means of ordering certain activities, of clarifying the effects of certain actions, and of clarifying the conditions under which those activities are valid.

Unless otherwise specified, at the moment of their creation or of DLE activation,

— all variables shall be initialized to their default value, or to their minimum permitted value if no default is specified;

— all counters shall be initialized to zero;

— all timers shall be initialized to inactive; and

— all queues shall be initialized to empty.

DL-Management may change the values of configuration variables.

5.7.1 Parameters, variables, counters, timers and queues to support the Basic class

The parameters, variables, counters, timers and queues defined in 5.7.1.1 through 5.7.1.18 are required in all DLEs. The variables, counters and timers defined in 5.7.1.19 through 5.7.1.27 are required in all DLEs except those with a time synchronism class (see 11.3(a)) of NONE.

5.7.1.1 V(ST) slot-time

V(ST) is used by the DLE to record the link's slot-time (see 3.3.25), which is a configuration parameter of the local link. Its range is 1 to 4095, and its unit is the transmission duration of one octet.

5.7.1.2 V(PhLO) per-DLPDU-PhL-overhead

V(PhLO) is used by the DLE to account for the PhL-induced delay between the end of the last octet of one DLPDU as it appears on the link, and the beginning of the first octet of any other DLPDU as it appears on the link, measured in units of one octet-duration. Its range is 2 to 63.

5.7.1.3 V(MRD) maximum-response-delay

V(MRD) is used by the DLE to record the link's maximum-response-delay (see 3.3.26), which is a configuration parameter of the local link. The default value for this variable is 3. Its range is 1 to 11, and its unit is one slot-time.

NOTE — This unit is chosen so that all measurements of an inactive link are for multiples of one slot-time.

5.7.1.4 V(IRRD) immediate-response-recovery-delay

V(IRRD) is used by the DLE to record the link's immediate-response-recovery-delay (see 3.3.26), which is a computed parameter of the local link. The default value for this variable is V(MRD) + 1. Its range is 2 to 12, and its unit is one slot-time.

5.7.1.5 V(MRC) maximum-retry-count

V(MRC) is used by the DLE to record the link's maximum-retry-count, which limits the number of immediate retries of a transaction which are permitted. V(MRC) is a configuration parameter of the local link. Its default value is 0, meaning immediate retries are not permitted. Its range is 0 to 7.

5.7.1.6 V(NRC) network-repeat-count

V(NRC) is used by the DLE to record the extended link's network-repeat-count, which specifies the maximum number of deferred retries of a multi-link communication which should be attempted. V(NRC) is a configuration parameter of the extended link, based to some extent on the values of V(MRC) for the various local links of the extended link. Its default value is 1, meaning a single deferred retry should be attempted when other considerations do not intervene. Its range is 0 to 7.

5.7.1.7 V(NDL) network-DLPDU-lifetime

V(NDL) is used by the DLE to record the extended link's network-DLPDU-lifetime, which specifies the maximum period that any DLPDU can remain in transit within the extended link. Its range is 1 ms to 60 s, in units of 1 ms, and its default value is 30 s.

5.7.1.8 V(TN) this-node

V(TN) is used by the DLE to record the local node-id (see 3.3.30), which is a configuration parameter of the local DLE. Its default uninitialized value is zero, which does not permit the DLE to transmit. Its range is 0, and 10_{16} to FF_{16} (see Annex A/A.2.2).

5.7.1.9 V(TL) this-link

V(TL) is used by the DLE to record the local link-id (see 3.3.29), which is a configuration parameter of the local link. Its default uninitialized value is zero, which does not permit a bridge to forward to another link. Its range is 0, and 1000_{16} to $FEFF_{16}$ (see Annex A/A.2.1).

5.7.1.10 V(MEP) DL-MAC-address-embedding-prefix

V(MEP) is used by the DLE (see Annex A/A.4) to embed the extended-link's 32-bit address space within the 48-bit address space used by the ISO/IEC LAN (local area network) standards (ISO/IEC TR 18802). The default value for this variable is zero. Its range is 0 to $3FFF_{16}$.

5.7.1.11 C(RD) remaining-duration counter

C(RD) is used by the DLE to record the remaining duration of token delegation when using a token delegated by a PASS TOKEN (PT) or EXECUTE SEQUENCE (ES) DLPDU, or the remaining link capacity until the next scheduled activity when using a scheduler token during link maintenance activity. C(RD) is initialized on receipt of such a DLPDU from the DLPDU's DD-parameter:

a) DLEs which implement C(RD) as an actual counter shall decrement C(RD) at the rate of one count per octet of elapsed transmission capacity until it reaches the value zero, after which no decrementing shall occur.

b) DLEs which do not implement C(RD) as an actual counter may instead approximate C(RD) as follows: each time the DLE uses the token, it decreases the approximated C(RD) by a pessimistic estimate of the amount of link capacity required for that instance of token use.

c) DLEs which do not implement C(RD) as either an actual or an approximated counter may instead

1) limit their token use at each received PT DLPDU to a sequence of transactions which is guaranteed to complete within the duration specified in the DD-parameter of that received PT DLPDU;

NOTE — This might be accomplished by comparing the value of the received DD-parameter with the worst case duration of the transaction

2) compute the minimum duration of token delegation required by the DLE in the next PASS TOKEN (PT) DLPDU to be received by the DLE; and

3) convey this value to the LAS DLE in a REQUEST INTERVAL (RI) DLPDU.

Such DLEs may also use a token conveyed by an EXECUTE SEQUENCE (ES) DLPDU specifying RESTART, but shall immediately return without using any token conveyed by an ES DLPDU specifying CONTINUE.

5.7.1.12 V(MID) minimum-inter-DLPDU-delay

V(MID) is used to specify the minimum duration for an interval of non-transmission which a transmitting DLE shall provide after receiving, or transmitting, a DLPDU. This interval is measured in units of the transmission duration of one octet; the measurement interval begins with either

a) the receipt of a PH-DATA confirm primitive by the DLE which confirms a PH-DATA request primitive which specified END-OF-DATA-AND-ACTIVITY; or

b) the receipt of a PH-DATA indication primitive by the DLE specifying END-OF-DATA, END-OF-DATA-AND-ACTIVITY, or END-OF-ACTIVITY;

whichever occurs first, and ends with the submission of a PH-DATA request primitive by the DLE specifying start-of-activity. The default value for this variable is 0, and its range is 0 to the smaller of 120 or (V(MRD)-1)×V(ST).

5.7.1.13 T(IRRD) immediate-response-recovery-delay monitor

T(IRRD) is used by the DLE which initiates a two-DLPDU transaction to monitor the local link for an anticipated immediate reply DLPDU, and to reassert its own token as the active token on the link if no reply is detected.

5.7.1.14 V(RA) reply-address

V(RA) is used by a DLE to record the DL-address to which a reply token was last passed by a COMPEL ACKNOWLEDGMENT (CA), COMPEL DATA (CD), or EXCHANGE DATA (ED) DLPDU. It also can be used by the LAS DLE to record the DL-address to which a reply token was last passed by a PROBE NODE-ADDRESS (PN) DLPDU.

5.7.1.15 V(OTA) outstanding-transaction-array

V(OTA) is used by the DLE, in transactions requiring an immediate reply, to correlate an immediate or delayed reply DLPDU with the initiating DLS-user request. It is also used by the DLE to determine, for any given transaction index, whether that index is currently assigned by the DLE to an incomplete outstanding transaction. V(OTA) is an array with an index range of 0 .. F_{16}.

5.7.1.16 V(LTI) last-transaction-index

V(LTI) is used by the DLE, in transactions requiring an immediate reply, to ensure that a new transaction does not reuse the index of the immediately previous transaction initiated by the same DLE. Its range is 0 .. F_{16}, and its initial value is F_{16}.

5.7.1.17 Q(US) unscheduled-service queue

Q(US) is used by the DLE to manage

a) **references** to the DLE's DL-address user-request queues (Q_A(UR) for all *A* within the DLE (see 5.7.3.1), including

   1) **references** which signal the need to compel a DLSDU transmission from the correspondent peer or publisher DLCEP, when *A* is a peer or subscriber DLCEP-address;

   2) **references** which signal the need to compel an instance of the unitdata-exchange service with the specified DLSAP-address, when *A* is a DLSAP-address whose DL(SAP)-role is INITIATOR;

   3) **references** which signal the need to transmit the contents of a sending buffer bound to a local DLCEP, when *A* is a local DLCEP-address; and

4) **references** which signal the need to send a DLSDU from the sending queue bound to local DLCEP or DLSAP-address, when *A* is a local DLCEP-address or DLSAP-address;

b) **references** to locally-scheduled active sequences (see Annex B) resulting from DL-SCHEDULE-SEQUENCE requests (see 9.4.3.1); and

c) DT DLPDUs containing DLS-user-data which are delayed responses to received CD and ED DLPDUs, queued in support of the DL-UNITDATA-EXCHANGE service.

This queue is used upon receipt of a PASS TOKEN (PT) DLPDU addressed to the DLE (see 3.3.17 and 6.2.2.3). The structure of this queue is described in 3.3.34.

5.7.1.18 V(RID) random identifier

V(RID) is a randomly-chosen identifier used by the DLE, in responding to a Probe NODE-address (PN) DLPDU addressed to the DLE, to introduce a random element into the response, which in turn is used to enable the DLE to validate a subsequently-received node-activation SPDU addressed to the DLE.

Each time it is used in a Probe Reply (PR) DLPDU, before its value is copied into the PR DLPDU, the variable shall be set to a new random value chosen uniformly from the range 0 to FF_{16}. The actual random choice shall be statistically independent of similar choices made by other DLEs.

NOTE — This requirement for statistical independence minimizes the probability of repeatedly-identical choices by identically-constructed real devices, and thus provides the basis for eventual discrimination among multiple DLEs which happen to respond to the same NODE DL-address, either through fault or misconfiguration.

5.7.1.19 C(NT) node-time counter

C(NT) is used by the DLE to record the monotonically-increasing component of the DLE's local node-time (see 3.3.24). In the absence of any corrective frequency adjustment (see 9.4.1.3), its nominal long-term rate of counting shall result in incrementing 8 192 000 units/s (= $2^{13} \times 10^3$ units/s). Its initial value shall be zero. C(NT) shall be sized to not roll-over before reinitialization.

Implementation NOTE — When other data is lacking, implementors may assume that the maximum period between reinitializations in industrial environments is five (5) years. The specified rate of counting does not imply the granularity of the counting, which depends upon the implementation and the time synchronization capability of the DLE.

5.7.1.20 V(LSTO) local-link-scheduling-time-offset

V(LSTO) is used by the DLE to record the signed offset (difference) between the sum N(NT) + N(LSTO) of the local LAS's local-link-scheduling-time, as received in a TIME DISTRIBUTION (TD) DLPDU, and the receiving DLE's C(NT), so that V(LSTO) = N_{LAS}(LSTO) + N_{LAS}(NT) – C(NT). The initial value of V(LSTO) shall be zero.

5.7.1.21 V(DLTO) DL-time-offset

V(DLTO) is used by the DLE to record the signed offset (difference) between the DL-time and the local-link-scheduling-time, such that DL-time = C(NT) + V(LSTO) + V(DLTO). This variable is also received in a TIME DISTRIBUTION (TD) DLPDU. Its initial value shall be zero.

5.7.1.22 V(TQ) time-quality

V(TQ) is used by the DLE to record the multi-partite quality of both the DLE and the source and path of time distribution on the extended link. Its initial value shall indicate a local time source.

5.7.1.23 V(MD) measured-delay

V(MD) is used by the DLE to record the filtered measured delay in two-way communications between the DLE and the current LAS, as measured with a series of ROUND-TRIP-DELAY QUERY/ROUND-TRIP-DELAY

REPLY DLPDUs. Its initial value, and its value when the DLE's own LAS functions are active, shall be zero. Its value is invalidated under the conditions specified in 9.4.1.3.

5.7.1.24 V(LN) LAS-node

V(LN) is used by the DLE to record the node-id (see 3.3.30) of the local link's LAS, as received in a TD (TIME DISTRIBUTION) DLPDU (see 9.4.1.3), or zero if the local link's LAS' node-id is unknown. Its initial value shall be zero. V(LN) is used to determine the validity and relevancy of the computed value of V(MD). Its range is the same as that of V(TN) (see 5.7.1.8).

5.7.1.25 V(TSC) time-synchronization-class

V(TSC) is used by the DLE to record the link's time-synchronization-class and to determine the DLE's own requirement for the minimum time distribution period which the DLE requires to maintain the specified level of time synchronization. V(TSC) shall specify one of the time-synchronization-classes defined in 11.3a. Its default value shall be the 10 ms time-synchronization class (see 11.3(a)).

5.7.1.26 T(TDP) time-distribution-period monitor

T(TDP) is used by the DLE to measure the time elapsed since the last sent or received TIME DISTRIBUTION (TD) DLPDU.

5.7.1.27 V(TSL) time-source-link

V(TSL) is used by the DLE to record the link-id of the source of time distribution on the extended link. Its range is the same as that of V(TL). Its initial value shall be zero.

5.7.2 Parameters and timers to support a DLS-user's request

Each specific instance of a DLS-user service request or response may have associated parameters and timers, depending on the type of request or response.

5.7.2.1 P_U(SDUL) DLSDU-length request parameter

P_U(SDUL) designates the length of the DLSDU associated with the specified DLS-user request or response.

5.7.2.2 P_U(SDU) DLSDU request parameter

P_U(SDU) designates the DLSDU associated with the specified DLS-user request or response. P_U(SDU) is conceptualized as an array of P_U(SDUL) octets, indexed from 1 to P_U(SDUL).

5.7.2.3 P_U(MCD) maximum-confirm-delay parameter

P_U(MCD) designates a DLS-user-established maximum confirm delay for a specific request, where the class of delay is always determined by associated text within this Part of this International Standard.

5.7.2.4 T_U(MCD) maximum-confirm-delay monitor

T_U(MCD) is used by the DLE to monitor the completion of the actions associated with the specified DLS-user request, to ensure that a corresponding DLS confirm is given to the DLS-user within the interval of time allotted by the DLS-user, as specified in the DLS-user-established maximum confirm delay for the specific request.

5.7.3 Queues to support DL-address-based DL-scheduling

Each specific instance of a DLSAP-address or DLCEP-address or subscriber DLCEP within the DLE shall have an associated user-request queue (see 3.3.33 and 5.7.3.1).

5.7.3.1 Q_A(UR) user-request queue

Each specific instance of Q_A(UR) is used by the DLE to manage DLS-user requests and responses which require DLPDU transmissions originating at the associated DL-address or subscriber DLCEP. The structure of this queue is described in 3.3.33.

5.7.4 Variables and timers to support a DLCEP

The state of each DLCEP is maintained in variables and timers specific to that DLCEP. The specific set of variables and timers required for a given DLCEP is dependent on the DLCEP's class and data delivery features.

As shown in Figure 3, the following relationships exist among the variables L (see 5.7.4.10) and H (see 5.7.4.11) and the negotiated receiving window size W_R at a receiving DLCEP, and the variables N (see 5.7.4.3), R (see 5.7.4.4), A (see 5.7.4.5), M (see 5.7.4.6), and the negotiated sending window size W_S (see 5.4.7.2) at the corresponding sending DLCEP:

If "SC"-subscripts designate a sending DLE of a DLC, and "RC"-subscripts designate a receiving DLE of the same DLC, then the relationships among the actual DLCEP variables, $V_C(\ldots)$, and negotiated parameters, $P_C(\ldots)$, from the point of view of a single direction of data transfer, are

$$P_{RC}(W_{RC}) = P_{SC}(W_{SC}) = W$$

$$V_{RC}(H) - W \le V_{SC}(M) - W \le V_{SC}(A) \le V_{RC}(L) \le V_{RC}(H) \le V_{SC}(M) \le V_{SC}(A) + W \le V_{RC}(L) + W$$

$$V_{SC}(M) - W \le V_{SC}(R) \le V_{SC}(M) < V_{SC}(N)$$

$$V_{RC,K}(RRS) \subseteq V_{RC,K}(MRS)$$

5.7.4.1 V_C(ST) DLCEP state

V_C(ST) is used by a DLE to maintain the current user state of the DLCEP. These states are the same as those shown in 9.2 Figure 7.

$$W_R = W_S = W$$

$$H - W \le M - W \le A \le L \le H \le M \le A + W \le L + W$$

$$M - W \le R \le M < N$$

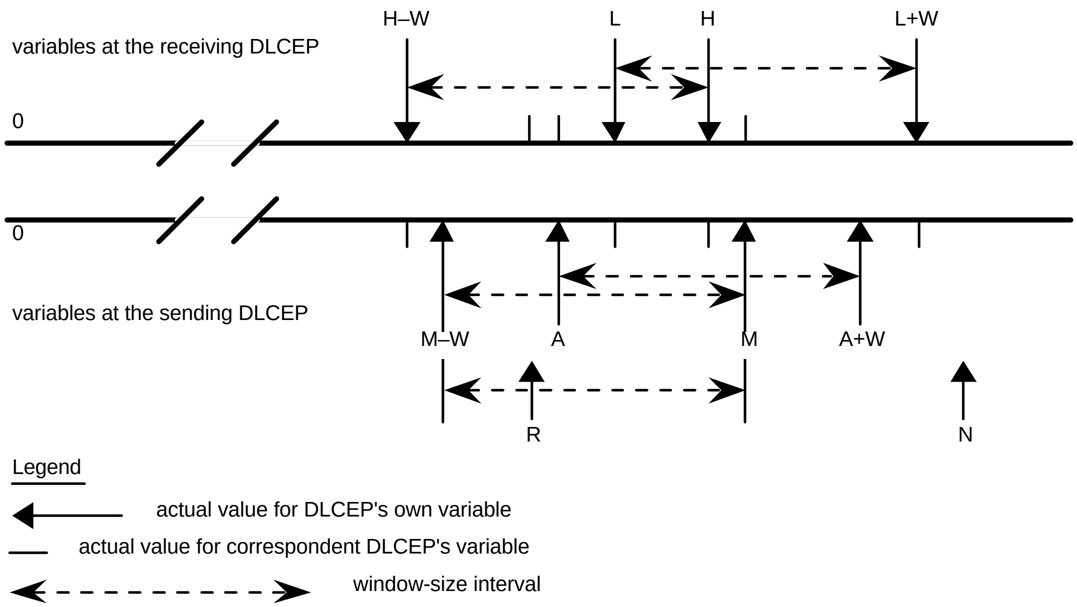


Figure 3 — Linear relationships of sending and receiving DLCEP sequence-number variables

5.7.4.2 V_C(NP) negotiated DLCEP parameters

V_C(NP) is used by a DLE to maintain the current DLCEP parameters as a single structured variable, both during and after the DLCEP-establishment process. These parameters are defined by the EC-parameters (see 8.1) of the ESTABLISH CONNECTION (EC) DLPDU.

Within this part of this International Standard, references to specific parameters within the EC parameters are denoted by the syntax "V.F," where V is a V_C(NP) variable and F is a field of the EC parameters as labeled in 8.1.

5.7.4.3 V_C(N) next sequence number to assign to a DLSDU

V_C(N) is used by a DLE to maintain the ordinal number one greater than the number of DLSDUs whose transmission on the DLCEP has been requested by the DLS-user. Its value at DLCEP establishment is one, and it increases monotonically each time the transmission of a new DLSDU is requested by the DLS-user. It does not increase when a DLS-user's request for transmission from a buffer (which must be on a DLCEP whose sending data delivery features are UNORDERED or ORDERED) results in the retransmission of a previously-transmitted DLSDU.

Implementation NOTE — This last requirement may be realized by a two-phase incrementation process, in which writing to the buffer arms the incrementation process for all DLCEPs which send from the buffer, and incrementation for a given DLCEP occurs during DLPDU formation immediately prior to transmission (but not retransmission) of the first segment of the buffer on that DLCEP.

5.7.4.4 V_C(R) maximum non-transmittable DLSDU sequence number

V_C(R) is used by a DLE for a DISORDERED or CLASSICAL PUBLISHER DLCEP, and optionally for an ORDERED PEER or PUBLISHER DLCEP, to maintain the ordinal number of the highest-numbered DLSDU which is no longer available for DLE transmission or retransmission, or zero when there is no such DLSDU. Its value at DLCEP establishment is zero, and it is non-decreasing. It increments each time a DLSDU, for which transmission was requested, becomes unavailable to the DLE for retransmission, either because the DLSDU has been queued within the DLL for longer than its specified maximum confirm delay, or because the DLSDU's sequence number differs from the highest sequence number transmitted by the sending window size $V_C(W_S)$, or due to DLC reset.

NOTE — Incrementing $V_C(R)$ may cause a corresponding number of requests to move from the third to the second partition of the relevant $Q_A(UR)$.

5.7.4.5 $V_C(A)$ maximum acknowledged DLSDU sequence number

$V_C(A)$ is used by a DLE for a DISORDERED or CLASSICAL PEER DLCEP to maintain the ordinal number of the highest-numbered DLSDU transmitted from this DLCEP which has been acknowledged by the correspondent DLE. Its value at DLCEP establishment is zero, and it is non-decreasing. A newly-received CA, CD, DT, ED or RC DLPDU may cause it to be increased.

NOTE — Incrementing $V_C(A)$ may cause a corresponding number of requests to move from the third to the second partition of the relevant $Q_A(UR)$.

5.7.4.6 $V_C(M)$ maximum transmitted DLSDU sequence number

$V_C(M)$ is used by a DLE for a sending DLCEP to maintain the ordinal number of the highest-numbered DLSDU which has at least one segment which is being, or has been, transmitted from this DLCEP. Its value at DLCEP establishment is zero, and it increases monotonically during the initial transmission of the first segment of each DLSDU.

5.7.4.7 $V_C(MS)$ maximum transmitted DLSDU segment number

$V_C(MS)$ is used by a DLE for an ORDERED, DISORDERED or CLASSICAL sending DLCEP to maintain the zero-based ordinal number of the highest-numbered segment of the $V_C(M)$'th DLSDU which is being, or has been, transmitted from this DLCEP. Its initial value at DLCEP establishment, before any DLSDU has been transmitted, is zero.

5.7.4.8 $V_{C,K}(SS)$ segments to send

$V_{C,K}(SS)$ is used by a DLE for a sending DLCEP to maintain the set of segments of the *K*'th DLSDU which still need to be transmitted. For each new (*N*'th) DLSDU presented by the DLS-user for transmission, a new list $V_{C,N}(SS)$ is created.

For $A<K\leq M$ and $R<K<N$, additions are made to the list upon receipt of a retransmission request for a segment of the *K*'th DLSDU, or upon timeout of the corresponding timer $T_{C,K}(SS)$ (or upon timeout of the timer $T_C(SS)$ when it applies to the *K*th DLSDU, when the permission of 5.7.4.9.1 is employed).

NOTE — The lowest-numbered segment of the lowest-numbered DLSDU remaining is usually the next to be transmitted; the DLE deletes this segment from the list upon transmission. For ORDERED or PUBLISHER sending DLCEPs, the first untransmitted segment of the first not-completely-transmitted DLSDU may be sent instead, to maintain an assured rate of delivery of new DLSDUs on periodic DLCs.

5.7.4.9 $T_{C,K}(SS)$ sent-segments monitor

$T_{C,K}(SS)$ is used by a DLE for a DISORDERED or CLASSICAL PEER DLCEP to monitor timely peer acknowledgment of the *K*'th transmitted DLSDU, where $A<K\leq M$. It is started when the DLSDU has been transmitted in its entirety, and is canceled when the DLSDU is acknowledged. Upon its timeout, the last segment of the *K*'th DLSDU is again added to $V_{C,K}(SS)$, causing its retransmission and potentially triggering the receiving DLE to request retransmission of any lower-numbered segments which also have been missed.

5.7.4.9.1 T_C(SS) simplified sent-segments monitor

It is permitted to replace the set of per-DLSDU timers { $T_{C,K}$(SS) } with a single per-DLCEP-timer, T_C(SS). Such a replacement will lead to equivalent DL-service when the number of outstanding DLSDUs is limited by other constraints, such as a small negotiated window size, or when the residual error rate of the communications path is very low. When such constraints do not apply, then this replacement may lead to inferior DL-service.

If such a replacement is made, then the following definition applies:

T_C(SS) is used by a DLE for a DISORDERED or CLASSICAL PEER DLCEP to monitor timely peer acknowledgment of transmitted DLSDUs. It is started, but not restarted, when a DLSDU has been transmitted in its entirety, and is canceled when a DLSDU is acknowledged. It is then restarted if there are still unacknowledged DLSDUs.

Upon its timeout, the last segment of the unacknowledged $DLSDU_K$ with the lowest sequence number is again added to the $V_{C,K}$(SS) corresponding to that DLSDU, causing its retransmission and potentially triggering the receiving DLE to request retransmission of any lower-numbered segments which also have been missed.

5.7.4.10 V_C(L) last-reported DLSDU sequence number

V_C(L) is used by a DLE for an ORDERED, DISORDERED or CLASSICAL receiving DLCEP to maintain the ordinal number of the last (highest-numbered) DLSDU received at the DLCEP and reported to the remote DLCEP. Its value at DLCEP establishment is zero, and it increases monotonically with the reporting of each completely-received DLSDU to the DLS-user.

5.7.4.11 V_C(H) highest-detected DLSDU sequence number

V_C(H) is used by a DLE for a PEER or SUBSCRIBER DLCEP to maintain the ordinal number of the highest-numbered DLSDU received, or detected as missing, at the DLCEP. Its value at DLCEP establishment is zero, and it increases with the reception of the first-received segment of each sequentially-higher-numbered newly-received DLSDU, or with the detection of a sequentially-higher-numbered missing DLSDU.

5.7.4.12 V_C(HS) highest-detected segment number of the highest-detected DLSDU sequence number

V_C(HS) is used by a DLE for a PEER or SUBSCRIBER DLCEP to maintain the zero-based ordinal number of the highest-numbered segment of the V_C(H)'th DLSDU which has been received, or detected as missing, at the DLCEP. Its value at DLCEP establishment is zero:

a) It is set to the zero-origin index of each newly-received segment, N_R(ASN) (see 8.4.2.1.(C.6)), received in a DLPDU provided that

   1) the received DLSDU number is greater than the prior value of V_C(H); or

   2) the received DLSDU number is the same as the prior value of V_C(H), and the received segment number is greater than the prior value of V_C(HS).

b) It is set to zero when V_C(H) is advanced upon detection of a missing DLSDU, and the number of segments in that DLSDU is unknown.

5.7.4.13 $V_{C,K}$(MRS) missing received segments

$V_{C,K}$(MRS) is used by a DLE for an ORDERED, DISORDERED or CLASSICAL PEER or SUBSCRIBER DLCEP to maintain the set of missing segments for the K'th DLSDU, for $L<K\leq H$. Additions are made to this list, or

one or more new lists created, upon receipt of any segment of a new DLSDU. Deletions are made as segments included on the list are received.

NOTES

1. When at least one segment of the *K*'th DLSDU has been received, then the total number of segments in the DLSDU is known and the missing segments list is exact. Otherwise it may be assumed that the missing DLSDU consists of only a single segment whose retransmission needs to be requested; upon receipt of any segment of the DLSDU, the list of segments still missing can be made exact.

2. Subscribers of a common publisher DLCEP are permitted, but are not required, to eavesdrop on each other's transmissions to the publisher. Such eavesdropping may provide early detection of message loss. Thus one or more new lists may also be created upon receipt of a request for retransmission of a segment of the *K*'th DLSDU, where *K*>H.

5.7.4.14 $V_{C,K}$(RRS) retransmission-request required segments

$V_{C,K}$(RRS) is used by a DLE for an ORDERED, DISORDERED or CLASSICAL, PEER or SUBSCRIBER DLCEP to maintain the set of missing segments for the *K*'th DLSDU for which retransmission requests are required, for L<*K*≤H. Additions are made to this list when a received DLPDU implies that one or more DLSDU segments or complete DLSDUs have been missed, and (for SUBSCRIBER DLCEPs) upon timeout of the timer $T_{C,K}$(RRS) used by the DLE to monitor for a successful response to a previous retransmission request for one or more segments of the same DLSDU.

Deletions are made upon reception (including reception during transmission) of a DLPDU which either contains, or requests the retransmission of, a missing segment.

NOTE — Such deletions upon reception of a request for retransmission for another subscriber on the same multi-peer DLC are optional (see 5.7.4.13/NOTE 2).

5.7.4.15 $T_{C,K}$(RRS) retransmission request monitor

$T_{C,K}$(RRS) is used by a DLE for a DISORDERED or CLASSICAL SUBSCRIBER DLCEP, and optionally for an ORDERED PEER or SUBSCRIBER DLCEP, to monitor for a successful response to a previous retransmission request for one or more segments of the *K*'th DLSDU, where L<*K*£H. The timer is started when not running, but not restarted while running, when a retransmission request is sent for one of the segments of the incomplete DLSDU. It is canceled when all segments of the DLSDU have been received. Upon its timeout, $V_{C,K}$(RRS) is set equal to the then-current value of $V_{C,K}$(MRS).

NOTE — The timer may also be started, but not restarted, when a retransmission request from another subscriber of the same multi-peer DLC, for one of the segments of the incomplete DLSDU, is overheard (see 5.7.4.13/NOTE 2).

5.7.4.16 T_C(RAS) residual activity stimulus

T_C(RAS) shall be used by a DLE at an ORDERED, DISORDERED or CLASSICAL, PUBLISHER or sending PEER DLCEP, when DLC establishment (see 9.2.1.1) requested residual activity on the DLC in this (sender-to-receiver) direction of data transfer, to ensure activity from that DLCEP during periods when all DLSDUs presented for transmission at that DLCEP since DLC establishment, or since the most recent DLC reset, have been transmitted

a) at a sending DISORDERED or CLASSICAL PEER DLCEP, when all of those DLSDUs have been acknowledged (that is, $V_C(A) = V_C(M)$); and

b) at a PUBLISHER DLCEP, or sending ORDERED PEER DLCEP, when none of those DLSDUs need to be (wholly or partially) retransmitted.

This timer restarts at every transmission from the DLCEP. At timeout, it causes the current state of the DLCEP to be sent to the correspondent DLCEP(s).

5.7.4.17 T_C(RAM) residual activity monitor

T_C(RAM) is used by a DLE for an ORDERED, DISORDERED or CLASSICAL, SUBSCRIBER or receiving PEER DLCEP to detect inactivity on the DLC. It is only required when DLC establishment (see 9.2.1.1) requested residual activity on the DLC in this (receiver-from-sender) direction of data transfer. It runs continuously, and restarts upon reception of any DLPDU on the DLCEP. At timeout, it causes a DLCEP (and possibly DLC) reset, which may in turn lead to a DLCEP (and possibly DLC) disconnect.

5.7.4.18 V_C(TNA) DL-time of last network access

V_C(TNA) is used by the DLE to record the DL-time at which a DL-BUFFER-RECEIVED indication or DL-BUFFER-SENT indication was last generated at the associated DLCEP. It is used only when the DLCEP has been specified as a synchronizing DLCEP during the establishment of one or more other local DLCEPs.

5.7.4.19 V_B(TW) DL-time of last buffer write

V_B(TW) is used by the DLE to record the DL-time at which all of a DLSDU was last written to the associated buffer (by a DL-PUT request primitive, or by the completion of reception into the buffer at a DLCEP). It is used only when the DLCEP establishment specified a DL-timeliness class other than NONE.

5.7.4.20 V_B(TP) DL-time of production

V_B(TP) is used by the DLE to record the DL-time at which a DLS-user transferred the associated DLSDU to the DLE. This is the DL-time at which the sending DLS-user issued the DL-PUT request which placed the associated DLSDU in the sending DL-buffer. It is used only when the DLCEP establishment specified a DL-timeliness class other than NONE.

5.7.4.21 V_B(TS) Timeliness-status of buffer-writing

V_B(TS) is used by the DLE to record the timeliness-status associated with the DLSDU stored in a buffer. If the buffer is written as a result of a DL-PUT request, then the value of this variable shall be equal to the DLS-user-data-timeliness. If the buffer is written as a result of reception of a DLPDU, then the value of this variable shall be updated as part of the reception procedure.

5.7.5 Variables and timers to support the Link Master class

The variables defined in 5.7.5 are required in all Link Master and Bridge DLEs.

5.7.5.1 V(DTA) delegation-address

V(DTA) is used by the LAS DLE to record the DL-address to which the token was last passed by a PASS TOKEN (PT), EXECUTE SEQUENCE (ES), or TRANSFER LAS (TL) DLPDU.

5.7.5.2 V(LL) local-link live-list

V(LL) is used by the LAS DLE to record the set of NODE DL-addresses (see 3.3.17 and 6.2.2.3) on the local link which appear to be in use by peer DLEs. It is used and updated by the LAS DLE during the LAS's address-probing and first-opportunity activities, as described in 10.2.

NOTE — Some previously-existing national standards refer to such a set as a “live list,” because it indicates the current set of interacting DLEs on the local link.

5.7.5.3 V(TCL) token-circulation list

V(TCL) is used by the LAS DLE to record the set of NODE DL-addresses (see 3.3.17 and 6.2.2.3) on the local link to which a PT DLPDU should be circulated to provide an opportunity for unscheduled transmissions, including requests to the LAS DLE for scheduling communications. This token circulation emulates the distributed token passing of prior national and international standards.

NOTE — This variable is used to restrict the circulation of an unrestricted token to only the “master” (that is, self-sufficient) nodes of the local link. This functionality assists in migration from some previously-existing national standards.

V(TCL) is used and updated by the LAS DLE during the LAS’s address-probing and first-opportunity activities, as described in 10.2. The DLEs represented in V(TCL) shall always be a subset of the DLEs represented in V(LL).

5.7.5.4 V(ENRL) expected-non-response list

V(ENRL) is used by the LAS DLE to record the set of NODE DL-addresses (see 3.3.17 and 6.2.2.3) on the local link of fractional-duty-cycle (FDC) DLEs which are expected to be non-responsive (that is, “asleep”). This is a subset of the set of FDC DLEs on the link, and changes dynamically as FDC DLEs inform the LAS DLE of:

a) their attentiveness (“wakefulness”); or

b) their intent to become non-responsive (to “go to sleep”).

Failure of a DLE listed in V(ENRL) to respond to a PT DLPDU addressed to the DLE, or an ES DLPDU addressed to a hierarchically-structured (see 6.2.2.1 format 1 or 6.2.2.2 format 1) DLSEP of the DLE,

1) shall not be treated as an error; and

2) the transmission of the PT or ES DLPDU shall not be retried.

NOTE — This last requirement is equivalent to using a value of zero (0) for V(MRC) when sending a PT DLPDU (see 7.15.3) to such a DLE.

5.7.5.5 V(MST) maximum-scheduled-traffic

V(MST) is used by the LAS DLE, during any dynamic schedule construction, to determine the maximum fraction of the theoretic interval V(TTRT) (see 5.7.5.11) which may be dedicated to explicitly scheduled DL-activity.

V(MST) is a configuration parameter of the local link which takes the form of a one-octet binary fraction. The range for this parameter is $0{,}00_{16}$ to $0{,}BF_{16}$; its default value is $0{,}40_{16}$.

5.7.5.6 V(MSO) maximum-scheduling-overhead

V(MSO) specifies the maximum scheduling overhead permitted an LAS DLE by the existing link schedule. Its range is 0 to $3F_{16}$; its default value is $3F_{16}$, and its unit is the transmission duration of one octet. This overhead is included in the time allocated for each scheduled activity, and so is used only during schedule construction and determination of whether a DLE can serve as LAS for an existing schedule.

5.7.5.7 V(DMDT) default-minimum-token-delegation-time

V(DMDT) is used by the LAS DLE to determine the default minimum amount of local link capacity which the LAS shall allocate to a DLE in a single PT DLPDU sent to the DLE.

V(DMDT) is a configuration parameter of the local link. The range for this variable is 20_{16} to $7FFF_{16}$, its default value is 54_{16} + V(PhLO), which permits the sending of one URGENT DLPDU, and its unit is the transmission duration of one octet.

5.7.5.8 V(DTHT) default-token-holding-time

V(DTHT) is used by the LAS DLE to specify the default initial amount of local link capacity which should be allocated to each DLE, in one cycle of “circulating the token,” by the LAS when the LAS sends one or more PT DLPDUs to the DLE.

V(DTHT) is a configuration parameter of the local link. The range for this variable is 114_{16} to 65 000, its default value is 114_{16} + V(PhLO), which permits the sending of one TIME-AVAILABLE DLPDU, and its unit is the transmission duration of one octet.

5.7.5.9 V(LTHT) link-maintenance-token-holding-time

V(LTHT) is used by the LAS DLE to specify the initial amount of local link capacity which should be allocated to LAS-related link maintenance activities in one cycle of "circulating the token." These activities include

— probing a single unused NODE DL-address via a PN DLPDU for the appearance of a DLE not currently active on the link, and the consequent sending of a DT DLPDU containing a node-activation SPDU to any such DLE;

— measuring the round-trip communications delay with another DLE on the local link via an RQ DLPDU;

— conveying updated schedule information to other Link Master DLEs on the local link to provide for schedule continuation after transfer of the LAS role to such a DLE; and

— coordinating a multi-link schedule with other LAS DLEs of the extended link.

V(LTHT) is a configuration parameter of the local link. The range for this variable is 124_{16} to 65 000, and its unit is the transmission duration of one octet. Its default value is

$$124_{16} + (3 \times \text{V(PhLO)}) + (\text{V(IRRD)} \times \text{V(ST)}),$$

which permits the probing of one NODE DL-address and the sending of one TIME-AVAILABLE DT DLPDU.

5.7.5.10 V(MTHA) maximum-token-holding-time-array

V(MTHA) is used by the LAS to specify, separately for each DLE listed in V(TCL) (see 5.7.5.3), the initial amount of local link capacity which should be allocated to that DLE, in one cycle of "circulating the token," by the LAS when the LAS sends one or more PT DLPDUs to that DLE. The range and units of this parameter are the same as V(DTHT) (see 5.7.5.8). Its default value is also V(DTHT).

5.7.5.11 V(TTRT) target-token-rotation-time

V(TTRT) is used by the LAS DLE to specify the desired upper bound on the time required for one cycle of "circulating the token" to all of the DLEs on the local link, as measured by the interval between successive occurrences of the LAS DLE sending a PT DLPDU, with a token-use subfield specifying RESTART (see 7.15.2), to the lowest-numbered NODE DL-address represented in V(LL).

V(TTRT) is a configuration parameter of the local link. The range for this variable is 1 to 60 000, its default value is 60 000, and its unit is one ms.

5.7.5.12 V(ATRT) actual-token-rotation-time

V(ATRT) is used by the LAS DLE to determine the actual time used for each cycle of "circulating the token" to all of the DLEs on the local link, as measured by the interval between successive occurrences of the LAS DLE's sending the PT DLPDU, with a token-use subfield specifying RESTART (see 7.15.2), to the lowest-numbered NODE DL-address represented in V(LL).

5.7.5.13 V(RTHA) remaining-token-holding-time-array

V(RTHA) is used by the LAS to specify, separately for each DLE listed in V(TCL) (see 5.7.5.3), the remaining amount of local link capacity which should be allocated to that DLE, in the current cycle of "circulating the token," by the LAS when the LAS sends one or more PT DLPDUs to that DLE. The range and units of this array are the same as V(MTHA), and it is reinitialized from V(MTHA) at the beginning of each cycle of "circulating the token."

5.7.5.14 V(NTHN) next-token-holding-node

V(NTHN) is used to record the NODE DL-address of the next DLE to which "the token should be circulated," which is the next NODE DL-address to which a PT DLPDU will be sent. Its range is 10_{16} to FF_{16} (see Annex A/A.2.2), and it advances cyclically through the set of DLEs specified by V(TCL) (see 5.7.5.3).

5.7.5.15 V(FUN) first-unpolled-node

V(FUN) specifies the first NODE DL-address of a series of consecutive NODE DL-addresses which are to be omitted from the orderly probe of NODE DL-addresses for DLEs not specified by V(LL). Its range is 14_{16} to $F7_{16}$ (see Annex A/A.2.2).

5.7.5.16 V(NUN) number-of-consecutive-unpolled-nodes

V(NUN) specifies the number of consecutive NODE DL-addresses which are to be omitted from the orderly probe of NODE DL-addresses for DLEs not specified by V(LL). Its range is 00_{16} to $E4_{16}$, and its default value is 0.

5.7.5.17 P(TRD) token-recovery-delay

P(TRD) is used by the LAS DLE to record the DLE's token-recovery-delay (see 3.3.28), which is determined solely by the DLE itself. The default value for this variable is 14. Its range is V(MRD)+3 to 14, and its unit is one slot-time.

NOTE — This unit is chosen so that all measurements of an inactive link are for multiples of one slot-time.

5.7.5.18 V(TDP) time-distribution-period

V(TDP) is used to determine the minimum frequency of time distribution on the local link. Its initial value shall be the minimum value required for the link's time-synchronization-class, V(TSC) as specified in 11.3(a). Its range is 5 ms to 55 s, and its unit is 1 ms.

5.7.5.19 V(MICD) maximum-inactivity-to-claim-LAS-delay

V(MICD) is used by the LAS DLE to record the link's maximum-inactivity-to-claim-LAS-delay (see 3.3.25.1), which is a configuration parameter of the local link. Its range is 1 to 4095, and its unit is one-eighth of the transmission period of one octet.

5.7.5.20 V(LDDP) LAS-data-base-distribution-period

V(LDDP) is used by the LAS to determine the time between two successive distributions of the LAS' data-base by means of LAS-data-base-status SPDUs sent on the local link. The range of V(LDDP) is 100 ms to 55 s, and its unit is 1 ms. The default value of this variable is 5 s.

5.7.6 Variables and timers to support the Bridge class.

The variables and timers defined in 5.7.6 and in Annex C are required in all Bridge DLEs.

5.7.6.1 V(ML) maximum-link

V(ML) may be used by a bridge DLE to limit the number of entries in the bridge's table of default forwarding information, which can have one entry for each link of the extended link. Its range is 1000_{16} to $FEFF_{16}$ (see Annex A/A.2.1), which is the maximum link-id defined for the extended network.

Additional variables and timers are defined in Annex C, mostly by reference to ISO/IEC 10038.

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

Within this clause, any reference to bit K of an octet is a reference to the bit whose weight in a one-octet unsigned integer is 2^K.

NOTE — This is sometimes referred to as "little endian" bit numbering.

6.1 PhIDU structure and encoding

Each PhIDU consists of Ph-interface-control-information (PhICI) and in some cases one octet of Ph-interface-data (see 5.4). When the DLE transmits a DLPDU, it computes a frame check sequence for the DLPDU as specified in 6.1.1, concatenates the DLPDU and 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, and 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 first to last octet of the DLPDU, and, after each Ph-DATA request primitive, awaits the consequent Ph-DATA confirm primitive.

c) The DLE issues a sequence of two 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.

d) The DLE issues a single Ph-DATA request primitive with PhICI specifying END-OF-DATA-AND-ACTIVITY, and awaits the consequent Ph-DATA confirm primitive.

The DLE forms a received DLPDU by concatenating the sequence of octets received as Ph-interface-control-information of consecutive Ph-DATA indications, computing a frame check sequence for those received octets as specified in 6.1.1, and checks the syndrome of the computed FCS for correctness as follows:

e) The DLE receives a single Ph-DATA indication primitive with PhICI specifying START-OF-ACTIVITY, and initializes its computation of an FCS for the received DLPDU.

f) 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 but the last two of those received octets to form the received DLPDU.

g) The DLE receives a single Ph-DATA indication primitive with PhICI specifying either END-OF-DATA, END-OF-DATA-AND-ACTIVITY or END-OF-ACTIVITY, and checks the syndrome of the computed FCS for correctness:

1) If the PhICI specified END-OF-DATA or END-OF-DATA-AND-ACTIVITY, and the computed FCS syndrome was correct, then the DLE reports the reconstructed DLPDU and the two octets of received FCS as a correctly-received DLPDU suitable for further analysis.

2) Otherwise the DLE increments its management statistics to reflect the erroneously-received DLPDU.

6.1.1 Frame check sequence

NOTE — The generator polynomial for this FCS is specified in Eq. 7. The polynomial for the receiver's expected residue is specified in Eq. 13. Exemplary implementations are discussed in NOTES, and are shown in Annex G.

In this part of this International Standard, as in other International Standards (for example, ISO 3309, ISO/IEC 8802-3, -4, -5, -6, and ISO/IEC 9314-2), DLPDU-level error detection is provided by calculating and appending a multi-bit Frame Check Sequence (FCS) to the other DLPDU fields during transmission to form a "systematic code word"1 of length ***n*** consisting of ***k*** DLPDU message bits followed by ***n - k*** redundant bits, and by calculating during reception that the message and concatenated FCS form a legal (***n***,***k***) code word. The value of ***n - k*** for this part of this International Standard is 16. The mechanism for this checking is as follows:

6.1.1.1 At the sending DLE

The original message (that is, the DLPDU without an FCS), the FCS, and the composite message code word (the concatenated DLPDU and FCS) shall be regarded as vectors M(X), F(X), and D(X), of dimension ***k***, ***n - k***, and ***n***, respectively, in an extension field over GF(2). If the message bits are $m_1 \ldots m_k$ and the FCS bits are $f_{n-k-1} \ldots f_0$, where $m_1 \ldots m_8$ form the first octet sent, $m_{8N-7} \ldots m_{8N}$ form the *N*th octet sent, and $f_7 \ldots f_0$ form the last octet sent, then the message vector M(X) shall be regarded to be

$$M(X) = m_1X^{k-1} + m_2X^{k-2} + \ldots + m_{k-1}X^1 + m_k \qquad \text{(Eq. 4)}$$

and the FCS vector F(X) shall be regarded to be

$$F(X) = f_{n-k-1}X^{n-k-1} + \ldots + f_0 \qquad \text{(Eq. 5)}$$

$$= f_{15}X^{15} + \ldots + f_0$$

The composite vector D(X), for the complete DLPDU, shall be constructed as the concatenation of the message and FCS vectors:

$$D(X) = M(X)\,X^{n-k} + F(X) \qquad \text{(Eq. 6)}$$

$$= m_1X^{n-1} + m_2X^{n-2} + \ldots + m_kX^{n-k} + f_{n-k-1}X^{n-k-1} + \ldots + f_0$$

$$= m_1X^{n-1} + m_2X^{n-2} + \ldots + m_kX^{16} + f_{15}X^{15} + \ldots + f_0$$

The DLPDU presented to the PhL shall consist of an octet sequence in the specified order.

The redundant check bits $f_{n-k-1} \ldots f_0$ of the FCS shall be the coefficients of the remainder F(X), after division by G(X), of $L(X)\,(X^k + 1) + M(X)\,X^{n-k}$

where G(X) is the degree ***n-k*** generator polynomial for the code words

$$G(X) = X^{n-k} + g_{n-k-1}X^{n-k-1} + \ldots + 1 \qquad \text{(Eq. 7)}$$

$$= X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^6 + X^3 + X^2 + X + 1$$

and L(X) is the maximal weight (all ones) polynomial of degree ***n-k-****1*

$$L(X) = = X^{n-k-1} + X^{n-k-2} + \ldots + X + 1 \qquad \text{(Eq. 8)}$$

1 W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes* (2nd edition), MIT Press, Cambridge, 1972.

$$= X^{15} + X^{14} + X^{13} + X^{12} + \ldots + X^2 + X + 1$$

That is,

$$F(X) = L(X)(X^k + 1) + M(X) X^{n-k} \pmod{G(X)} \qquad \text{(Eq. 9)}$$

NOTES

1. The L(X) terms are included in the computation to detect initial or terminal message truncation or extension by adding a length-dependent factor to the FCS.

2. This G(X) polynomial is relatively prime to all, and is thus not compromised by any, of the polynomials commonly used in DCEs (modems): the differential encoding polynomial $1 + X^{-1}$ and all primitive scrambling polynomials of the form $1 + X^{-j} + X^{-k}$.

3. Code words D(X) constructed from this G(X) polynomial have Hamming distance 4 for lengths ≤ 344 octets and Hamming distance 5 for lengths ≤ 15 octets.

4. As a typical implementation, at a transmitter, the initial remainder of the division is preset to all ones. The transmitted message bit stream is multiplied by X^{16} and divided (modulo 2) by the generator polynomial G(X), specified in Eq. 7. The ones complement of the resulting remainder is transmitted as the 16-bit FCS.

6.1.1.2 At the receiving DLE

The octet sequence indicated by the PhE shall be concatenated into the received DLPDU and FCS, and regarded as a vector V(X) of dimension ***u***

$$V(X) = v_1 X^{u-1} + v_2 X^{u-2} + \ldots + v_{u-1} X + v_u \qquad \text{(Eq. 10)}$$

NOTE 1 — Because of errors **u** can be different than **n**, the dimension of the transmitted code vector.

A remainder R(X) shall be computed for V(X), the received DLPDU and FCS, by a method similar to that used by the sending DLE (see 6.1.1.1) in computing F(X):

$$R(X) = L(X) X^u + V(X) X^{n-k} \pmod{G(X)} \qquad \text{(Eq.11)}$$

$$= r_{n-k-1} X^{n-k-1} + \ldots + r_0$$

Define E(X) to be the error code vector of the additive (modulo-2) differences between the transmitted code vector D(X) and the received vector V(X) resulting from errors encountered (in the PhS provider and in bridges) between sending and receiving DLEs.

$$E(X) = D(X) + V(X) \qquad \text{(Eq. 12)}$$

If no error has occurred, so that E(X) = 0, then R(X) will equal a non-zero constant remainder polynomial

$$R_{ok}(X) = L(X) X^{n-k} \pmod{G(X)} \qquad \text{(Eq.13)}$$

$$= X^{15} + X^{14} + X^{13} + X^9 + X^8 + X^7 + X^4 + X^2$$

whose value is independent of D(X). Unfortunately R(X) will also equal $R_{ok}(X)$ in those cases where E(X) is an exact non-zero multiple of G(X), in which case there are “undetectable” errors. In all other cases, R(X) will not equal $R_{ok}(X)$; such DLPDUs are erroneous and shall be discarded without further analysis.

NOTE 2 — As a typical implementation at a receiver, the initial remainder of the division is preset to all ones. The received bit stream is multiplied by X^{16} and divided (modulo 2) by the generator polynomial G(X), specified in Eq. 7. The resulting 16-bit remainder should be 1110 0011 1001 0100 (X^{15} through X^0, respectively) in the absence of errors.

6.1.1.3 Modification within bridges

When forwarding a DLPDU, it is sometimes necessary for a bridge to alter one or more subfields of a DLPDU's frame control field. When making these modifications, the bridge shall modify the received FCS to compensate for changes in the frame control octet; the bridge shall not discard the received FCS and recompute a new FCS after the DLPDU's frame control field has been altered2.

When the received DLPDU's length, plus that of its FCS field, is N octets, then the bridge can compensate for a change in bit K in the first octet by computing the residual of the polynomial

$$X^{8N+K-8} \quad (\text{modulo } G(X)) \qquad \text{(Eq. 14)}$$

and then updating the DLPDU's FCS field by exclusive-ORing the computed residual into that field.

Implementation NOTE — When the bridge initializes, it can precompute the residuals for all permissible DLPDU lengths and bit positions potentially needing alteration— for values of N between 3 and 272, and for K equal to 2. Then for any DLPDU the bridge need only apply to the DLPDU's FCS that residual which corresponds to the change actually made in the DLPDU's frame control octet.

6.2 Common DLPDU structure, encoding and elements of procedure

Each DLPDU consists of a frame control field which specifies the type of DLPDU and conveys small size (fractional octet) parameters of the DLPDU; zero to three explicit address fields, each containing a DL-address, all of the same length; additional parameters of the DLPDU; and for most DLPDUs, a user data field conveying all or part of a DLSDU. To this is appended before transmission, and removed after reception, an FCS field (see 6.1.1) used to check the integrity of the received DLPDU.

6.2.1 Frame control (FC) field

The frame control (FC) field consists of one octet. It specifies the type of the DLPDU. For many types of DLPDU it also conveys a number of fractional-octet parameters, known as frame-control subfields, specific to the DLPDU type.

Some types of DLPDU require an immediate reply. Such DLPDUs may only be sent while holding a scheduler or delegated token.

When a token is delegated, then the priority specified in the token DLPDU is the minimum priority required of all DLPDUs sent during the following period of token use. DLPDUs of lower priority shall not be sent during that period.

6.2.1.1 Address size subfield

An address-size subfield is used to specify the number of octets in each address field of the DLPDU.

In DLPDUs which support multiple address sizes, this subfield occupies bit 3 of the frame control field. Its encoding is:

0) **SHORT** — the DLPDU's address fields are each two octets;

---

2 D. R. Irvin, *Preserving the integrity of cyclic-redundancy checks when protected text is intentionally altered*, IBM Journal of Research and Development, Vol. 33, No. 6, November 1989, pp. 618-626

1) **LONG** — the DLPDU's address fields are each four octets, and their link designator subfields are non-zero if possible.

The DT DLPDU also has a special form with only implicit DL-addresses:

— **VERY-SHORT** — the DLPDU logically contains a single DL-address field, which is null (zero octets in length).

i) If the single null DL-address field is a source address, then the source DL-address of the current DLPDU is implicitly the destination DL-address of the immediately-prior CA, CD or ED DLPDU on the link (which must have been a DLCEP-address).

ii) If the single null DL-address field is a destination address, then the destination DL-address of the current DLPDU is implicitly the source DL-address of the immediately-prior CA, CD or ED DLPDU on the link (which itself may be an implicit DLSAP- or DLCEP-address).

NOTE — This VERY SHORT form is indicated by other components of the DLPDU's frame control octet, and not by its address-size subfield.

6.2.1.2 Final token use subfield

A final-token-use designator is used to optimize the return of a delegated token at the end of the last transaction of an instance of delegated token usage. This subfield is present in most DLPDUs, including all DLPDUs which can be sent by a DLE which holds a delegated token or reply token. When present, this subfield occupies bit 2 of the frame control field. Its encoding and semantics depend on the type of DLPDU:

a) The encoding and semantics for CA, CD, CT, ED and RQ DLPDUs is:

0: **NOT-FINAL** — the delegated token is not returned to the LAS at the end of the current transaction;

1: **FINAL** — the delegated token is returned to the LAS at the end of the current transaction, after transmission of the DLPDU and its requested immediate reply DLPDU, whether detected or not, and no additional use of the token being returned is needed at this time.

b) The encoding and semantics for DC, DT, EC, RC, RR, SR and TD DLPDUs is:

0: **NOT-FINAL** — the delegated token is not returned to the LAS at the end of the current transaction;

1: **FINAL** — the delegated token is returned to the LAS at the end of the current transaction, after transmission of the DLPDU, and no additional use of the token being returned is needed at this time.

c) The encoding and semantics for PT and ES DLPDUs is:

0: **RESTART** — this is the initial token delegation within the current cycle of "circulating the token" or of scheduled sequence execution, and so the indicated sequence and any repetitively-scheduled transactions should be restarted;

1: **CONTINUE** — this is a subsequent (that is, secondary) token delegation within the current cycle of "circulating the token" or of scheduled sequence execution, and so the previously-initiated sequence of queued or scheduled transactions should be continued.

When a bridge receives for forwarding a DLPDU which contains a final-token-use subfield whose value is different than required by the bridge, then the bridge shall complement the received DLPDU's final-token-use subfield before forwarding and shall make a compensating modification to the received DLPDU's FCS field (see 6.1.1.3) to preserve the FCS integrity protection provided by the DLPDU's originator.

NOTE — Only CA, CD, DC, DT, EC, ED and RC DLPDUs (see 7.1 through 7.7) are forwarded through bridges.

6.2.1.3 Priority subfield

A Priority designator is used to specify the DLPDU's or transaction's priority, to limit the size of the user data field of the DLPDU, and to constrain the minimum (lowest) priority of any DLPDU sent as an immediate reply requested by, or token usage delegated by, this DLPDU. This subfield is present in all CA, CD, DT, ED, PT and RC DLPDUs which contain explicit DL-addresses. When present, this subfield occupies bits 1 and 0 of the frame control field. Its encoding is:

01: **URGENT** (high) priority;

10: **NORMAL** (medium) priority;

11: **TIME-AVAILABLE** (low) priority.

6.2.2 DL-address fields

The structure of DL-addresses is specified in Annex A, which also specifies standard pre-assigned addresses and address ranges.

6.2.2.1 LONG address field

A **LONG** address field usually consists of a fixed sequence of three parts as specified in Annex A. The parts are

a) an explicit link designator component;

b) an explicit node designator component; and

c) an explicit selector component.

Separately, the three parts reflect an hierarchical address structure. This hierarchy may be partially or totally flattened. These alternatives are shown in Figure 4.

format 1

Link	Node	Selector
2 octets	1 octet	1 octet

or

format 2

Link	Node || Selector
2 octets	2 octets

or

format 3

Link || Node || Selector
4 octets

Figure 4 — DL-address alternative structures

When present, the Link field shall be delocalized on transmission and reception, as specified in 6.2.2.4. Similarly, when the Link field is present and specifies the local link, and when the Node field is present, then the Node field shall be delocalized on transmission.

6.2.2.2 SHORT address field

A **SHORT** address field usually consists of a fixed sequence of one implicit and two explicit parts, as specified in Annex A and Figure 5. The parts are

a) an implicit link designator component, specifying the local link, which is always present;

b) an explicit node designator component; and

c) an explicit selector component.

Separately, the parts reflect an hierarchical address structure. This hierarchy may be partially flattened. The alternatives are shown in Figure 5.

format 1	**0000**	**Node**	**Selector**
		1 octet	1 octet

or

format 2	**0000**	**Node || Selector**
		2 octets

Figure 5 — SHORT DL-address field — alternative implicit structures

When present, the Node field shall always be delocalized on transmission, as specified in 6.2.2.4.

6.2.2.3 NODE DL-address field

A **NODE** DL-address is a one-octet version of a SHORT address field which designates some DLE's DL-support functions. It consists of a fixed sequence of two implicit and one explicit parts, as specified in Annex A and Figure 6. The parts are

a) an implicit link designator component, specifying the local link;

b) an explicit node designator component, specifying the DLE; and

c) an implicit selector component of zero, specifying the DLE's DL-support functions.

format 1	**0000**	**Node**	**00**
		1 octet	

Figure 6 — NODE DL-address field — implicit structure

The Node field shall always be delocalized on transmission, as specified in 6.2.2.4.

6.2.2.4 Delocalization

The following transmission and reception processes are collectively referred to as delocalization (see 3.3.31, which is the DLE-internal process of converting synonymous DL-addresses to a canonical form for transmission, or for DL-address recognition during reception.

Two values of the Link field — the value zero and the value equal to V(TL) — are equivalent in designating the local link:

a) On transmission, when the value of an explicit Link field would otherwise be zero, then the value of that Link field shall always be set to V(TL).

NOTES

1. The variable V(TL) can have the value zero, in which case this substitution effects no change.

2. This equivalence makes it possible for a DLE to transmit on the local link without knowing the link's correct value for V(TL), or during periods when that value is in transition (in which case it is being administratively changed from zero to non-zero).

3. This equivalence provision is an aid to DLS-users, and potentially in implementations of this DL-protocol; it permits references to the local link DL-address component to be represented uniformly by the value zero.

b) On reception, with respect to the Link field, the value V(TL) shall be considered equivalent to the value zero, except that a DLPDU containing a DL-address with a Link field actually equal to zero shall not be forwarded by a bridge onto a different link.

Within a DLE, and when addressing the local link, two values of the Node field — the value zero and the value equal to V(TN) — are equivalent in designating the local DLE on the local link.

c) On transmission, when the value of the Link field is equivalent to V(TL), and the value of a Node field would otherwise be zero, then the value of that Node field shall always be set to V(TN).

NOTE — This equivalence provision is an aid to DLS-users, and potentially in implementations of this DL-protocol; it permits references to the local node DL-address component to be represented uniformly by the value zero.

6.2.3 Parameter field

Each DLPDU class may have a DLPDU-class-specific parameter field; these are all described in 8.

When it is desirable to distinguish between the values of a DLE variable, V(xx), or counter, C(xx), which is copied into a parameter field, and the current value of the same variable or counter, then the value within the parameter field is referred to as N(xx), because it no longer tracks changes in the value of the source variable or counter. The need for this separation of nomenclature is particularly apparent in the case of counters such as C(NT), which never stop counting, and in the various time-related DLPDUs (TD, RQ and RR), which may contain multiple fields based on various samplings of C(NT).

6.2.4 User data field

RC, CA, DT and ED DLPDUs, associated with the connectionless and connection-oriented data transfer services, contain a user data field which is used to convey either a partial or complete DLSDU from one DLS-user to another. The size of this user data field is constrained to be no larger than that permitted by the priority of the conveying DLPDU:

01: **URGENT** (high) priority: ≤ 64 octets;

10: **NORMAL** (medium) priority: ≤ 128 octets;

11: **TIME-AVAILABLE** (low) priority: ≤ 256 octets.

PR DLPDUs contain a user data field which is used to convey a probe-response SPDU (see Annex B/ B.3.2.1) from the token-holding DLE to the current LAS DLE; its size is constrained to be no larger than permitted at URGENT priority — 64 octets.

TL DLPDUs contain a user data field which is used to convey a LAS-data-base-status SPDU (see Annex B/ B.3.2.3) from the LAS DLE to the addressed DLE; its size is constrained to be no larger than permitted at URGENT priority — 64 octets.

DC and EC DLPDUs contain a user data field. Their implicit priority is TIME-AVAILABLE; however, the size of their data fields is constrained to be no larger than permitted at NORMAL priority — 128 octets.

6.2.5 Elements of procedure for minimum-inter-DLPDU delay

The DLE which holds the dominant token shall start transmission only after providing a delay of at least minimum-inter-DLPDU delay, V(MID), octet durations, where the delay is measured as specified in 5.7.1.12.

6.2.6 Elements of procedure for dropping of token by dominant token holder

If the DLE which holds the dominant token has just completed either

a) the transmission of a DLPDU, as indicated by receipt of a Ph-DATA confirm primitive corresponding to the most-recent Ph-DATA request primitive, which shall have specified END-OF-DATA-AND-ACTIVITY; or

b) the reception of a DLPDU, as indicated by receipt of a Ph-Data indication primitive specifying either end-of-data-and-activity or end-of-data or end-of-activity

but has not yet started its next transmission, either

c) because it is waiting for the required interval of minimum-inter-DLPDU delay; or

d) because it is not ready to transmit due to delay in preparing the next transmission;

then if that DLE is able to receive during this interval and if a PhL-indication (see 5.4.3) reporting DATA is received, then the DLE shall drop the token.

6.2.7 Common elements of procedure for monitoring link activity

A number of DLPDU types (CA, CD, ED, CT, RQ, PT, ES, TL) sent by a token holder request an immediate response. Common procedures apply for those DLEs which

a) need to monitor for that immediate response; and

b) need to determine

— when to stop monitoring for that response; and

— whether or not that response occurred.

6.2.7.1 Monitoring for an immediate response by the initiating DLE

After sending a CA, CD, ED, CT or RQ, DLPDU which requests an immediate response,

a) where the sending DLE is not also the responding DLE, then the sending DLE shall

1) monitor the local link for a period of immediate-response-recovery-delay slot-times, V(IRRD) $\times$ V(ST) octet-durations, waiting for a reply; and

2) then take appropriate action based on the result of that monitoring;

b) where the sending DLE is also the responding DLE, it shall take action as if the monitoring had occurred and had successfully detected the requested immediate response DLPDU.

The monitoring procedure is:

1) If a PhL-indication (see 5.4.3) reporting DATA is received, then the DLE shall

i) stop further monitoring;

ii) then wait for receipt of the PhL-indication reporting END-OF-DATA-AND-ACTIVITY or END-OF-ACTIVITY;

2) If 1) does not apply, and the monitoring period expires, then

i) if the bus is not active at that moment (that is, the last-received PhL-indication reported END-OF-ACTIVITY), then the DLE shall stop further monitoring.

ii) if i) does not apply, implying that the link is active at that moment (that is, the last-received PhL-indication reported START-OF-ACTIVITY), then

α) Unless (β) applies, the DLE shall monitor the local link for a period of one additional slot-time, V(ST) octet-durations, waiting for a PhL-indication:

A) If a PhL-indication reporting DATA is received, then the DLE shall proceed as in (1).

B) If a PhL-indication reporting END-OF-ACTIVITY is received, and (A) does not apply, then the DLE shall stop further monitoring.

C) If neither (A) nor (B) apply, and the monitoring period expires before a PhL-indication is received, then the DLE shall wait for receipt of the PhL-indication reporting END-OF-DATA-AND-ACTIVITY or END-OF-ACTIVITY.

β) Implementations based on hardware that was designed specifically to implement this International Standard, where such hardware was demonstrable on or before 31 December 1995 alternatively may just proceed as in (α.C) without any additional period of monitoring.

NOTE — After this document achieved initial ACDV status, implementors were encouraged to develop chips to assist in evaluating this complex protocol. As a consequence, it was found desirable to improve the noise rejection characteristics of the token passing process, and the text (2.ii.α), which would require changes in those existing chips, was the result. (β) grandfathers the noise rejection approach of the first ACDV for those early implementations, and only for those early implementations.

At the end of monitoring, the sending DLE shall act based on the result of that monitoring:

3) If (2.i) or (2.ii.α.B) applied, then the sending DLE shall

— report the failure to detect a DLPDU to local DL-management; and

— if the final-token-use subfield of the originating CA, CD, ED, CT or RQ DLPDU specified NOT-FINAL, then the DLE shall start the next transmission within immediate-response-recovery-delay plus one slot-times, $(V(IRRD) + 1) \times V(ST)$ octet-durations, of the end of transmission of that CA, CD, ED, CT or RQ DLPDU.

4) If (1) or (2.ii.α.A) or (2.ii.α.C) applied and link activity did not result in a DLPDU, then the sending DLE shall

— report the failure to detect a valid reply to local DL-management; and

— if the final-token-use subfield of the originating CA, CD, ED, CT or RQ DLPDU specified NOT-FINAL, then the DLE shall start the next transmission within maximum-reply-delay slot-times, $V(MRD) \times V(ST)$ octet-durations, of the beginning of the current period of link non-activity;

5) If

— the link activity resulted in a DLPDU; and

— if the received DLPDU was not a permissible reply DLPDU,

then any held token shall be dropped and local DL-management shall be notified of the event.

NOTE — These DL-management reports may take the form of incrementing a DL-management error counter.

6) If

— the link activity resulted in a DLPDU; and

— the received DLPDU was a permissible reply DLPDU; and

— the final-token-use subfield of the originating CA, CD, ED, CT or RQ DLPDU specified NOT-FINAL,

then the DLE shall start the next transmission within maximum-reply-delay slot-times, V(MRD) × V(ST) octet-durations, of the beginning of the current period of link non-activity.

6.2.7.2 Monitoring for an immediate response by the LAS DLE

If the final-token-use subfield of the received CA, CD, ED or RQ DLPDU has the value FINAL, then

a) if the LAS DLE is not the addressed immediate responder DLE, then the LAS DLE shall apply the monitoring procedures of 6.2.7.1(1) and 6.2.7.1(2); and

b) if the LAS DLE is also the responding DLE, then it shall take action as if the monitoring had occurred and had successfully detected the immediate response DLPDU.

After completing the monitoring (a), or after sending the requested immediate response (b), the LAS DLE shall

1) assume that the current use of the delegated token has terminated and that the scheduler token is again dominant on the local link; and

2) treat that termination as if the token had been returned by an RT (see 7.17) DLPDU; and

3) resume active operation as the LAS.

6.2.7.3 Monitoring for an immediate response as a subscriber to a DLC

A DLE which

— receives a CA, CD or ED DLPDU requesting an immediate response; and

— is an intended recipient of that requested immediate response DLPDU; and

— is not itself the CA, CD or ED DLPDU's addressed responder,

shall initiate a timer with a duration of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations:

a) If a PhL-indication (see 5.4.3) reporting DATA is received, then the DLE shall stop further monitoring and wait for receipt of the PhL-indication reporting END-OF-DATA-AND-ACTIVITY or END-OF-ACTIVITY and proceed as in (e).

b) If the monitoring period expires, and the bus is not active at the end of this monitoring period, then the DLE shall stop further monitoring and proceed as in (d).

c) If the monitoring period expires, and the bus is active at the end of this monitoring period, then the DLE shall wait for receipt of the PhL-indication reporting END-OF-DATA-AND-ACTIVITY or END-OF-ACTIVITY, and then proceed as in (d) or (e) as appropriate.

d) If the monitoring did not result in a DLPDU as in (b) or possibly in (c), then the DLE shall invalidate V(RA).

e) If the link activity resulted in a DLPDU as in (a) or possibly in (c), and if the received DLPDU was a DT DLPDU whose source DL-address is implicitly or explicitly the destination DLCEP-address specified by the CA, CD or ED DLPDU, then the recorded address V(RA) shall be assumed to be the implied source address of that CA, CD or ED DLPDU. Otherwise the recorded address V(RA) shall be invalidated.

6.2.8 Two-way alternate (half-duplex) operation

There is no requirement for a sending DLE to receive PH-DATA indications while it is transmitting a DLPDU; but when such indications are not receivable (that is, when the DLE is operating in a two-way alternate or half-duplex mode), then a sending DLE shall treat each transmitted DLPDU as if it had been received concurrently by that DLE.

7 DLPDU-specific structure, encoding and elements of procedure

This clause defines the structure, contents and encoding of each DLPDU type and format, and specifies elements of procedure for that DLPDU type and format. Table 2 summarizes their structure.)

Within each sub-clause, the DLPDU's structure, contents and encoding are described first. Then those aspects relating to the sending and receiving DLS-users and their DLEs are addressed, followed by those aspects, if any, that are unique to the store-and-forwarding functions of bridges (relay DLEs).

A DLE which is not ONLINE to the local link, but which is attempting to change its DL-state to ONLINE, may send a DLPDU only as a response to a PN DLPDU addressed to the DLE's DL-support functions, as specified in 7.13.4, 7.14.3 and 10.2.1. During this interval the DL-protocol specifies the response of the DLE to received PN, PT and DT DLPDUs addressed to the DLE's DL-support functions. The DLE's behavior after it changes its state to ONLINE shall not be dependent in any way on the receipt before it went ONLINE of other than the above-specified (PN, PT, DT) DLPDUs.

NOTE — Receipt of a DLPDU without observable side-effects is a purely local matter, and is thus outside the scope of standardization.

A DLE which is ONLINE to the local link may send a DLPDU only when it holds the dominant token on the local link. The set of DLPDUs which may be sent while holding the dominant token is dependent on the class of the dominant token, as summarized in Table 3:

a) If the dominant token is a scheduler token, then the permitted DLPDU classes are CA, CD, DC, DT, EC, ED, ES, PN, PT, RC, RQ, TD, TL, WK and IDLE.

b) If the dominant token is a delegated token, then the permitted DLPDU classes are CA, CD, CT, DC, DT, EC, ED, RC, RI, RQ, RT, WK and IDLE.

c) If the dominant token is a reply token, then the permitted DLPDU classes are

   1) DT and SR when replying to a CA, CD or ED DLPDU;

   2) TD when replying to a CT DLPDU;

   3) PR when replying to a PN DLPDU; and

   4) RR when replying to an RQ DLPDU.

d) Tokens are created only by the following DLPDUs:

   1) A scheduler token is created only by a CL DLPDU;

   2) A delegated token is created only by PT and ES DLPDUs; and

   3) A reply token is created only by CA, CD, CT, ED, PN, RQ and TL DLPDUs

.

Table 2 — Summary structure of DLPDUs

DLPDU class	frame control	DL-addresses destination	source	2nd source	parameters	user data
EC 1	1111 LF00	[HL.]N.S	[HL.]N.S	[HL.]N.S	EC-p	o-DLSDU
EC 2	1110 LF00		[HL.]N.S	[HL.]N.S	EC-p	o-DLSDU
DC 1	0111 LF00	[HL.]N.S	[HL.]N.S		DC-p	o-DLSDU
DC 2	0110 LF00		[HL.]N.S		DC-p	o-DLSDU
RC 1	0111 LFPP	[HL.]N.S	[HL.]N.S		RC-p	o-DLSDU
RC 2	0110 LFPP		[HL.]N.S		RC-p	o-DLSDU
CA 1	1110 LFPP	[HL.]N.S	[HL.]N.S		SD-p	o-pDLSDU
CA 2	1010 LFPP	[HL.]N.S	—		SD-p	o-pDLSDU
CD 1	1111 LFPP	[HL.]N.S	[HL.]N.S		o-SD-p	—
CD 2	1011 LFPP	[HL.]N.S	—		o-SD-p	—
ED 1	1100 LFPP	[HL.]N.S	[HL.]N.S		SD-p	pDLSDU
ED 2	1000 LFPP	[HL.]N.S	—		SD-p	pDLSDU
DT 1	1101 LFPP	[HL.]N.S	[HL.]N.S		SD-p	o-pDLSDU
DT 2	1001 LFPP	[HL.]N.S	—		SD-p	o-pDLSDU
DT 3	0101 LFPP		[HL.]N.S		SD-p	o-pDLSDU
DT 4	1001 0F00	[PSA]	—		SD-p	o-pDLSDU
DT 5	0101 0F00		[PDA]		SD-p	o-pDLSDU
SR	0001 0F11	[PSA]	N		o-SR-p	—
CT	0001 0F00	—	—		—	—
TD	0001 0F01	—	N		TD-p	—
RQ	1100 0F00	N.0	N.0		RQ-p	—
RR	1101 0F00	N.0	N.0		RR-p	—
PN	0010 0110	N	—		PN-p	—
PR	0010 0111	—	—		—	SPDU
PT	0011 0FPP	N	—		DD-p	—
ES	1000 LF00	[HL.]N.S	—		DD-p	—
RT	0011 0100	—	[DTH]		—	—
RI	0010 0000	—	[DTH]		DD-p	—
CL	0000 0001	—	N		—	—
TL	0000 0110	N	—		—	SPDU
WK	0000 0000	N	—		—	—
Idle	0001 0F10	—	—		—	o-DLSDU

LEGEND:
L indicates the length of the associated DL-addresses (0 = SHORT, 1 = LONG)
F indicates final use of a token, or that a sequence should be finished rather than restarted
PP specifies the priority of the DLPDU and any passed token
shading indicates a logically non-existent field
— indicates a logically existent field whose contents are required to be null
[HL.]N.S is a four-octet LONG DL-address (HLNS) when L = 1
or a two-octet SHORT DL-address (NS) with HL = 00 implied when L = 0
N is a one-octet NODE DL-address
N.0 is the two-octet SHORT DL-address form of a one-octet NODE DL-address
[PDA] is the implied DL-address equal to the explicit destination DL-address of the immediately prior DLPDU on the link, which must have been a CA, CD or ED DLPDU
[PSA] is the implied DL-address equal to the implied or explicit source DL-address of the immediately prior DLPDU on the link
o- indicates optional field contents
xx-p indicates xx-class DLPDU parameters
DLSDU is a DL Service Data Unit
pDLSDU is a complete or partial DLSDU
SPDU is a Support Protocol Data Unit

Table 3 — DLPDU restrictions based on dominant token

DLPDU class	Type of token created	Can be sent while using Scheduler token	Can be sent while using Delegated token	Can be sent in Reply to
EC	none	Y	Y	—
DC	none	Y	Y	—
RC	none	Y	Y	—
CA	Reply	Y	Y	—
CD	Reply	Y	Y	—
ED	Reply	Y	Y	—
DT	none	Y	Y	CA, CD, ED
SR	none	N	N	CA, CD, ED, TL
CT	Reply	N	Y	—
TD	none	Y	N	CT
RQ	Reply	Y	Y	—
RR	none	N	N	RQ
PN	Reply	Y	N	—
PR	none	N	N	PN
PT	Delegated	Y	N	—
ES	Delegated	Y	N	—
RT	none	N	Y	—
RI	none	N	Y	—
CL	none	N	N	—
TL	Scheduler	Y	N	—
WK	none	Y	Y	—
Idle	none	Y	Y	—

7.1 Establish Connection (EC) DLPDU

An ESTABLISH CONNECTION (EC) DLPDU is used to establish a peer DLC between two DLS-users, or a multi-peer DLC between a publishing DLS-user and subscribing DLS-users.

NOTE — Use of this DLPDU to modify the characteristics of an existing DLC are for future study.

7.1.1 Structure of the EC DLPDU

Table 4 — Structure of EC DLPDUs

format	frame control	destination address	source address	second source address	parameters	user data
1L	1111 1F00	HL.N.S	HL.N.S	HL.N.S	EC-p	o-DLSDU
1S	1111 0F00	N.S	N.S	N.S	EC-p	o-DLSDU
2L	1110 1F00		HL.N.S	HL.N.S	EC-p	o-DLSDU
2S	1110 0F00		N.S	N.S	EC-p	o-DLSDU

7.1.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is TIME-AVAILABLE;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU terminates use of a delegated token.

7.1.1.2 The address field shall consist of either

a) an explicit destination DL-address and two explicit source DL-addresses, in that order, for formats 1L and 1S; or

b) only two explicit source DL-addresses, for format 2L and 2S.

For formats 1L and 2L, all addresses shall be LONG; for formats 1S and 2S, all addresses shall be SHORT.

7.1.1.3 The establish-connection parameters (EC-parameters) field specifies the proposed or selected QoS attributes of the DLC, including the version of the DL protocol in use. This field shall be structured and encoded as described in 8.1.

7.1.1.4 The user data field shall consist of a single optional DLSDU whose maximum size is limited to 128 octets.

7.1.2 Content of the EC DLPDU

The frame control field shall be encoded as specified in Table 4.

For formats 1L and 1S,

a) the first DL-address shall be a DL(SAP)-address or a DLCEP-address; and

b) the second DL-address shall be a DLCEP-address or a DLSAP-address; and

c) the third DL-address shall be a DLSAP-address of the DLSAP associated with that second DL-address.

For formats 2L and 2S, the first DL-address shall be a publisher DLCEP-address, and the second DL-address shall be a DLSAP-address of the DLSAP associated with that first DL-address.

The EC-parameters shall specify the proposed or selected QoS attributes of the DLC, including the version of the DL-protocol in use. The contents of this field shall be as described in 8.1.

7.1.3 Sending the EC DLPDU

An EC DLPDU may be sent on the link when the sending DLE holds a scheduler token or delegated token which is the dominant token on the local link, and when the remaining allocated duration of token usage, C(RD), permits completion of the EC DLPDU's transmission prior to expiration of the token.

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) before transmission.

If the DLE holds a delegated token, and no additional use of that token after sending this DLPDU is needed at that time, then the DLE may set the final-token-use subfield of the EC DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

7.1.4 Receiving the EC DLPDU

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) upon reception.

A received EC DLPDU shall be treated as follows by the receiving DLE:

7.1.4.1 Actions required of all DLEs

If the first DL-address specified by the DLPDU designates an active DL(SAP)-address or an active DLCEP of the DLE, then the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.1.4) for further processing.

7.1.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.1.4.3 Additional actions required of a Bridge class DLE

If the first DL-address specified in the DLPDU is one which the bridge should forward, and the bridge was able to buffer the DLPDU without error, then the received DLPDU shall be forwarded with modification of the frame-control field in the forwarded DLPDU as appropriate (see 6.1.1.3).

The bridge shall attempt to update its routing table entry for the source DLCEP-address and source DLSAP-address specified in the DLPDU to reflect the bridge port from which the DLPDU was received.

If the DLPDU's Establish-Connection parameters indicate that the DLPDU's addressee(s) will be the subscriber(s) of a multi-peer DLC, then the bridge shall attempt to update its routing table entry for the source DLCEP-address specified in the DLPDU by adding the bridge port(s) to which the DLPDU is being forwarded to the set of sink DLCEP ports associated with that source DLCEP-address.

NOTE — This last procedure is only meaningful for formats 1L and 1S of the EC DLPDU.

7.1.4.4 Additional actions required of the current LAS DLE

If the final-token-use subfield of the received DLPDU has the value FINAL, then the LAS DLE shall

a) assume that the current use of the delegated token is terminated as if the token had been returned by an RT (see 7.17) DLPDU; and

b) assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.2 Disconnect Connection (DC) DLPDU

A DISCONNECT CONNECTION (DC) DLPDU is used to disconnect an existing or proposed DLC.

7.2.1 Structure of the DC DLPDU

Table 5 — Structure of DC DLPDUs

format	frame control	destination address	source address	parameters	user data
1L	0111 1F00	HL.N.S	HL.N.S	DC-p	o-DLSDU
1S	0111 0F00	N.S	N.S	DC-p	o-DLSDU
2L	0110 1F00		HL.N.S	DC-p	o-DLSDU
2S	0110 0F00		N.S	DC-p	o-DLSDU

7.2.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is TIME-AVAILABLE;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU terminates use of a delegated token.

7.2.1.2 The address field shall consist of either

a) an explicit destination DL-address and an explicit source DL-address, in that order, for formats 1L and 1S; or

b) only an explicit source DL-address, for format 2L and 2S.

For formats 1L and 2L, all addresses shall be LONG; for formats 1S and 2S, all addresses shall be SHORT.

7.2.1.3 The disconnect-connection parameters (DC-parameters) field shall specify the version of the DL protocol in use, the desired action and the reason for that action. This field shall be structured and encoded as described in 8.2.

7.2.1.4 The user data field shall consist of a single optional DLSDU whose maximum size is limited to 128 octets.

7.2.2 Content of the DC DLPDU

The frame control field shall be encoded as specified in Table 5.

For formats 1L and 1S, either

a) both DL-addresses shall be peer DLCEP-addresses; or

b) the first DL-address shall be a DL(SAP)-address, and the second DL-address shall be a DLCEP-address; or

c) the first DL-address shall be a DLCEP-address, and the second DL-address shall be a DLSAP-address.

For formats 2L and 2S, the sole DL-address shall be a publisher DLCEP-address.

The DC-parameters shall specify the version of the DL-protocol in use, the desired action and reason, and other information. The contents of this field shall be encoded as described in 8.2.

7.2.3 Sending the DC DLPDU

A DC DLPDU may be sent on the link when the sending DLE holds a scheduler token or delegated token which is the dominant token on the local link, and when the remaining allocated duration of token usage, C(RD), permits completion of the DLPDU's transmission prior to expiration of the token.

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) before transmission.

If the DLE holds a delegated token, and no additional use of that token after sending this DLPDU is needed at that time, then the DLE may set the final-token-use subfield of the DC DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

7.2.4 Receiving the DC DLPDU

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) upon reception.

A received DC DLPDU shall be treated as follows by the receiving DLE:

7.2.4.1 Actions required of all DLEs

If the first DL-address specified by the DLPDU designates an active DLSAP-address or an active DLCEP of the receiving DLE, then the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.1.7) for further processing.

7.2.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.2.4.3 Additional actions required of a Bridge class DLE

If the first DL-address specified in the DLPDU is one which the bridge should forward, and the bridge was able to buffer the DLPDU without error, then the received DLPDU shall be forwarded with modification of the frame-control field in the forwarded DLPDU as appropriate (see 6.1.1.3).

If the DLPDU's Disconnect-Connection parameters indicate that a reply is not requested, then the bridge may attempt to remove the routing table entries for any explicit DLCEP addresses specified in the DLPDU.

7.2.4.4 Additional actions required of the current LAS DLE

If the final-token-use subfield of the received DLPDU has the value FINAL, then the LAS DLE shall

a) assume that the current use of the delegated token is terminated as if the token had been returned by an RT (see 7.17) DLPDU; and

b) assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.3 Reset Connection (RC) DLPDU

A RESET CONNECTION (RC) DLPDU is used to reset an existing or proposed DLC.

7.3.1 Structure of the RC DLPDU

Table 6 — Structure of RC DLPDUs

format	frame control	destination address	source address	parameters	user data
1L	0111 1FPP	HL.N.S	HL.N.S	RC-p	o-DLSDU
1S	0111 0FPP	N.S	N.S	RC-p	o-DLSDU
2L	0110 1FPP		HL.N.S	RC-p	o-DLSDU
2S	0110 0FPP		N.S	RC-p	o-DLSDU

7.3.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's priority;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU terminates use of a delegated token.

7.3.1.2 The address field shall consist of either

a) an explicit destination DL-address and an explicit source DL-address, in that order, for formats 1L and 1S; or

b) only an explicit source DL-address, for format 2L and 2S.

For formats 1L and 2L, all addresses shall be LONG; for formats 1S and 2S, all addresses shall be SHORT.

7.3.1.3 The reset-connection parameters (RC-parameters) field shall specify the version of the DL protocol in use, the desired action and reason, and other information. This field shall be structured and encoded as described in 8.3.

7.3.1.4 The user data field shall consist of a single optional DLSDU whose maximum size is limited to the maximum DLS-user-data size permitted for a DLPDU of the priority specified in 7.3.1.1(b).

7.3.2 Content of the RC DLPDU

The frame control field shall be encoded as specified in Table 6.

For formats 1L and 1S, both DL-addresses shall be peer DLCEP-addresses. For formats 2L and 2S, the sole DL-address shall be a publisher DLCEP-address.

The RC-parameters shall specify the version of the DL protocol in use, the desired action and reason, and other information. The contents of this field shall be encoded as described in 8.3.

7.3.3 Sending the RC DLPDU

An RC DLPDU may be sent on the link when the sending DLE holds a scheduler token or delegated token which is the dominant token on the local link, and when the remaining allocated duration of token usage, C(RD), permits completion of the DLPDU's transmission prior to expiration of the token.

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) before transmission.

If the DLE holds a delegated token, and no additional use of that token after sending this DLPDU is needed at that time, then the DLE may set the final-token-use subfield of the RC DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

7.3.4 Receiving the RC DLPDU

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) upon reception.

A received RC DLPDU shall be treated as follows by the receiving DLE:

7.3.4.1 Actions required of all DLEs

If the first DL-address specified by the DLPDU designates an active DLCEP of the receiving DLE, then the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2) for further processing.

7.3.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.3.4.3 Additional actions required of a Bridge class DLE

If the receiving DLE is a bridge, and the first DL-address specified in the DLPDU is one which the bridge should forward, and the bridge was able to buffer the DLPDU without error, then the received DLPDU shall be forwarded with modification of the frame-control field in the forwarded DLPDU as appropriate (see 6.1.1.3).

7.3.4.4 Additional actions required of the current LAS DLE

If the final-token-use subfield of the received DLPDU has the value FINAL, then the LAS DLE shall

a) assume that the current use of the delegated token is terminated as if the token had been returned by an RT (see 7.17) DLPDU; and

b) assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.4 Compel Acknowledgment (CA) DLPDU

A COMPEL ACKNOWLEDGMENT (CA) DLPDU is used

a) to transfer (or retransfer) a limited amount of transparent user data from one requesting DLS-user to another DLS-user; and

b) to request that the transfer be acknowledged as soon as possible,

without requiring that the transaction (request and acknowledgment) occur within the context of a DLC. It is also used for similar purposes within the context of a DLC, and to assist in the synchronization of DLCEPs and of their DLS-users.

A CA DLPDU creates and passes a reply token to the addressed receiving DLE, which upon reception becomes the dominant token on its link. A CA DLPDU requires an immediate reply of either a DATA (DT) DLPDU with no DLS-user-data or a STATUS RESPONSE (SR) DLPDU. If no DLPDU is received in reply, then the transaction is repeated a maximum of V(MRC) times.

7.4.1 Structure of the CA DLPDUs

Table 7 — Structure of CA DLPDUs

format	frame control	destination address	source address	parameters	user data
1L	1110 1FPP	HL.N.S	HL.N.S	SD-p	o-pDLSDU
1S	1110 0FPP	N.S	N.S	SD-p	o-pDLSDU
2L	1010 1FPP	HL.N.S	—	SD-p	o-pDLSDU
2S	1010 0FPP	N.S	—	SD-p	o-pDLSDU

7.4.1.1 The frame control field shall specify

a) the DLPDU's function, including that an immediate reply is being requested;

b) the transaction's and DLPDU's priority;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU, and either its expected immediate reply or the appropriate link-idle timeout, terminates use of a delegated token.

NOTE — This field necessarily has the value NOT-FINAL when an immediate retry of the current transaction is possible. Only a transaction which is guaranteed to not need an immediate retry may have the value FINAL specified in its associated DLPDU.

7.4.1.2 The address field shall consist of either

a) an explicit destination DL-address and an explicit source DL-address, in that order, for formats 1L and 1S; or

b) only an explicit destination DL-address, for formats 2L and 2S.

For formats 1L and 2L, all addresses shall be LONG; for formats 1S and 2S, all addresses shall be SHORT.

7.4.1.3 The status-data-parameters (SD-parameters) field specifies information appropriate to the associated destination DL-address:

a) If that DL-address is a DLSAP-address, then the SD-parameters specify

1) a transaction-id used by the originating DLE to correlate a delayed returned reply with the originating request; and

2) a DLSDU-priority used to convey the actual priority of the accompanying DLSDU to the responding DLE, or that there is no accompanying DLSDU.

This field shall be structured and encoded as described in 8.4.1.

b) If that DL-address is a DLCEP-address, then the SD-parameters specify state information for the addressed DLCEP. This field shall be structured and encoded as described in 8.4.2.

NOTE — The size and structure of this field is dependent on the QoS attributes associated with the DLCEP addressed by the destination DL-address specified in this DLPDU, and is determined during DLCEP establishment.

7.4.1.4 The user data field size and content are limited by the associated destination DL-address:

a) If that DL-address is a DLSAP-address, then the user data field shall consist of DLS-user-data whose maximum size is limited to the smaller of the maximum DLS-user-data sizes permitted for a DLPDU of the priorities specified in 7.4.1.1(b) and 7.4.1.3(a.2), and shall not be null.

b) If that DL-address is a DLCEP-address, then the user data field shall consist of DLS-user-data whose maximum size is limited to the smaller of:

1) the maximum DLS-user-data size permitted for a DLPDU of the priority specified in 7.4.1.1(b); and

2) the maximum DLSDU size negotiated on the DLC for data transmission to that DLCEP, and may be null.

7.4.2 Content of the CA DLPDU

The frame control field shall be encoded as specified in Table 7.

Either the DL-addresses shall be

a) two explicit DLSAP-addresses; or

b) one explicit DLCEP-address, and a second explicit or implicit DLCEP-address; or

c) one explicit DLCEP-address, followed by one explicit or implicit DLSAP-address.

7.4.2.1 Content of the CA DLPDU when specifying a destination DLSAP-address

When the first DL-address is a DLSAP-address as in 7.4.2(a), then

a) if the DLPDU format is format 1L or 1S, then

   1) this DLPDU is being used to implement the unitdata transfer service with remote DLE confirmation;

   2) the DL(SAP)-role for the destination DLSAP-address shall be BASIC;

   3) the second address shall be present, shall be a DLSAP-address, and the DL(SAP)-role for that DLSAP-address shall be BASIC;

   4) the SD-parameters field shall specify a DLSDU-priority and a transaction-id used by the originating DLE to correlate a delayed returned reply with the originating request, where

      i) the contents of this field shall be as described in 8.4.1; and

      ii) the DLSDU-priority shall be the priority of the accompanying user data and shall be the same as the DLPDU-priority specified in 7.4.1.1(b); and

   5) the user data shall be a single DLSDU whose size is limited to the maximum size for the priority specified in 7.4.1.4(a), and shall not be null;

b) no other DLPDU format may be used.

7.4.2.2 Content of the CA DLPDU when specifying a destination DLCEP-address

When the first address is a DLCEP-address, as in 7.4.2(b) and 7.4.2(c), then

a) this DLPDU can convey a single or partial DLSDU

   — from one peer DLCEP to its corresponding peer DLCEP; or

   — from a subscriber DLCEP to its corresponding publisher DLCEP,

   shall request state information from the addressed DLCEP, and shall not permit DLS-user-data to be included in the reply DLPDU; and

b) the second address, if present,

   — shall be the peer DLCEP-address of the same DLC as the destination peer DLCEP-address; or

   — shall be a subscriber DLCEP's calling-DLSAP-address of the same DLC as the destination publisher DLCEP-address; and

c) the SD-parameters field shall specify state information for the addressed DLCEP, and the contents of this field shall be as described in 8.4.2; and

   NOTE 1 — The size and structure of this field is dependent on the QoS attributes associated with the DLCEP addressed by the destination DL-address specified in this DLPDU, and is determined during DLCEP establishment.

d) the user data shall specify those octets of a DLSDU consistent with the negotiated DLSDU size and the segmentation information specified in the accompanying SD-parameters, and may be null.

   NOTE 2 — A CA DLPDU with null user data can be used by a DLE with a subscriber DLCEP to solicit current DLC state information from the corresponding publisher DLCEP.

   NOTE 3 — Formats 1L and 1S are used for peer-to-peer (see 7.4.2(b)) and subscribers-to-publisher (see 7.4.2(c)) communications when the DLPDU-authentication attribute is SOURCE or MAXIMAL. Formats 2L and 2S are used for peer-to-peer and subscribers-to-publisher communications when the DLPDU-authentication attribute is ORDINARY.

It is a protocol error if the above conditions are not met.

7.4.3 Sending the CA DLPDU

A CA DLPDU may be selected for transmission on the link when

a) the sending DLE holds a scheduler token or delegated token which is the dominant token on the local link;

b) the remaining allocated duration of token usage, C(RD), permits completion of V(MRC)+1 implied transactions prior to expiration of the token, where each transaction consists of sending the CA DLPDU that requires an immediate reply, and awaiting a worst-case SR DLPDU or worst-case permitted DT reply DLPDU not containing DLS-user-data; and

c) if the CA DLPDU will be addressed to a DLSAP-address, then the outstanding-transaction-array, V(OTA) (see 5.7.1.15), searched circularly from the last-transaction-index, V(LTI) (see 5.7.1.16), has an unassigned entry whose index is not V(LTI).

Once selected, if the CA DLPDU will be addressed to a DLSAP-address, then

— V(LTI) shall be set to the index of that unassigned entry in V(OTA); and

— that entry in V(OTA) shall be assigned to the selected CA DLPDU and shall record information which permits an expected response DT DLPDU to be correlated with the specific invocation of the unitdata-transfer service which gave rise to the CA DLPDU.

Once selected, transmission of the CA DLPDU shall be retried until either

1) a permissible immediate reply DLPDU is received; or

2) an impermissible DLPDU is received when an immediate reply was expected; or

3) the original transmission and the permitted maximum number of transmission retries, V(MRC) (see 5.7.1.5), have all failed to elicit one of the permissible reply DLPDUs.

If the DLE holds a delegated token, and no additional use of that token after sending this DLPDU and awaiting its immediate reply is needed at that time, then the DLE may set the final-token-use subfield of the CA DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

NOTE – The FINAL transaction is necessarily the (V(MRC)+1)'th in a series; otherwise, if a reply is not received, then another CA DLPDU would have to be sent before the current cycle of token use was completed, making the current transaction NOT-FINAL.

Each explicit DL-address in the CA DLPDU shall be delocalized (see 6.2.2) before transmission.

After sending a CA DLPDU, the sending DLE shall monitor the local link for a reply as specified in 6.2.7.1. The permissible reply DLPDU is either

— a DT DLPDU whose destination DL-address is implicitly or explicitly the originating DLSAP-address or DLCEP specified by the CA DLPDU; or

— a DT DLPDU without a destination DL-address; or

— an SR DLPDU.

If V(LTI) was assigned to the transaction, as a result of the search described in (c), and if a permissible reply DLPDU was not received, then the V(LTI)'th entry shall be deassigned.

7.4.4 Receiving the CA DLPDU

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) upon reception.

A received CA DLPDU shall be treated as follows by the receiving DLE:

7.4.4.1 Actions required of all DLEs

NOTE — The next alternative attempts to detect the reception of a duplicated CA 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 reply DT DLPDU. In such a case the response DT DLPDU is required to be identical to the first in those fields and subfields which convey DLS-user-data and related information.

a) If

1) the destination DL-address specified by the DLPDU designates an active DLSAP-address of the receiving DLE, or an active DLCEP-address of a DLC for which the receiving DLE is peer or publisher;

2) the immediately-prior DLPDU was transmitted as an immediate-response to a received CA DLPDU whose destination DL-address was the same DLSAP-address or DLCEP-address;

3) if a source DL-address is present in the just-received CA DLPDU, then it was also present and identical in that prior received CA DLPDU;

4) starting with the SD-parameters field, the first three octets of the just-received CA DLPDU, or the remainder of the DLPDU if fewer than three octets, are identical to the corresponding octets of that prior received CA DLPDU; and

5) a period of link inactivity of (immediate-response-recovery-delay + 1) slot-times, $(V(IRRD) + 1) \times V(ST)$ octet-durations, has not occurred since receipt of that prior received CA DLPDU,

then the receiving DLE

i) shall discard the received DLPDU and not forward it to the DLE's upper-level functions for further processing; and

ii) shall retransmit the prior-transmitted immediate-reply DT DLPDU, within a period of maximum-response-delay slot-times, $V(MRD) \times V(ST)$ octet-durations, of receipt of the CA DLPDU, unchanged from the prior transmission except for those subfields of

— the final-token-use subfield of the frame control octet which conveys information to the listening LAS; and

— those subfields of any present DLCEP SD-parameters which acknowledge receipt of, or request retransmission of, received DLSDUs.

NOTE — This requirement is meant to ensure that the DL-priority, content and identity of any DLS-user-data conveyed in the immediate reply DT DLPDU is identical to that in the immediately-prior DT DLPDU sent from this same DL-address, including the originator's transaction-id if one is present in the received and transmitted SD-parameters.

b) If (a) does not apply, and the destination DL-address specified by the DLPDU designates a DLSAP-address of the receiving DLE, then the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.3) for further processing.

NOTE — Group DL-addresses are not covered here; such CA DLPDUs are erroneous and are unrecognized upon receipt.

The receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, $V(MRD) \times V(ST)$ octet-durations, of receipt of the CA DLPDU. The reply DLPDU shall be a DT DLPDU with any explicit destination DL-address in the DT DLPDU having the same value as the source DL-address of the received CA DLPDU, and length and format as follows:

— DT (see 7.7.1 format 2L) in response to CA (see 7.4.1 format 1L); or

— DT (see 7.7.1 format 4) in response to CA (see 7.4.1 format 1S),

and shall include SD-parameters specifying

— the originator's transaction-id as received in the stimulating CA DLPDU, as specified in 7.4.2.1(a.4); and

— reception status for the DLSDU conveyed by that CA DLPDU.

The reply DT DLPDU shall not contain any DLS-user-data.

c) If (a) does not apply, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE

— is a peer or publisher, and the DLL priority of the DLCEP is not equal to the priority specified in the received DLPDU; or

— is a peer, and the length and number of DL-addresses is not as expected or the DLPDU specifies an explicit source address which is not equal to the remote peer DLCEP's DLCEP-address,

then

1) the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing; and

2) the receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, $V(MRD) \times V(ST)$ octet-durations, of receipt of the CA DLPDU. The reply DLPDU shall be a DT DLPDU in the format negotiated for the DLC for the selected direction of transmission, and shall contain SD-parameters appropriate to the sending DLCEP, but shall not contain DLS-user-data.

d) If neither (a) nor (c) applies, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE is a peer or publisher, then

1) the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing;

2) the receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, $V(MRD) \times V(ST)$ octet-durations, of receipt of the CA DLPDU. The reply DLPDU shall be a DT DLPDU in the format negotiated for the DLC for the selected direction of transmission, shall contain SD-parameters appropriate to the sending DLCEP, but shall not contain DLS-user-data; and

3) the requesting CA DLPDU may contain information about the state of the requesting DLCEP. The reply DT DLPDU is permitted, but is not required, to reflect that state information in its reply; immediate processing of that state information before sending the immediate reply shall be permitted but shall not be required.

e) If (a) does not apply, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE is a subscriber, then

1) the non-DLS-user-data portion of the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing;

2) the receiving DLE shall record the destination DL-address from the received CA DLPDU in V(RA) for subsequent association with the expected immediate reply DT DLPDU, which should be the next DLPDU received; and

3) the receiving DLE shall monitor the local link for a reply and then act based on the result of that monitoring, all as specified in 6.2.7.3.

7.4.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.4.4.3 Additional actions required of a Bridge class DLE

If all of the following conditions hold

a) the destination DL-address specified in the DLPDU is one which the bridge should forward but to which the bridge DLE itself would not otherwise generate an immediate reply DLPDU;

b) the immediately-prior DLPDU was transmitted as an immediate-response to a received CA DLPDU with the same destination DL-address;

c) if a source DL-address is present in the just-received CA DLPDU, then it was also present and identical in that prior received CA DLPDU;

d) the basic-DLC-parameters portion of the SD-parameters of the just-received CA DLPDU are identical to those of that prior received CA DLPDU; and

e) a period of link inactivity of immediate-response-recovery-delay plus one slot-times,

$(V(IRRD)+1) \times V(ST)$ octet-durations, has not occurred since receipt of that prior received CA DLPDU,

then the bridge

1) shall discard the received DLPDU and not forward it to the bridge's other functions for further processing; and

2) shall initiate retransmission of the prior-transmitted immediate-reply SR DLPDU, unchanged from the prior transmission.

NOTE — This requirement is meant to ensure that any status conveyed in the immediate reply SR DLPDU is identical to that in the immediately-prior SR DLPDU.

Otherwise,

i) If the destination DL-address specified in the DLPDU is one which the bridge should forward but which the bridge DLE itself would not otherwise receive, then the bridge shall form and send an SR DLPDU

— within a period of maximum-response-delay slot-times, $V(MRD) \times V(ST)$ octet-durations, of receipt of the CA DLPDU; and

— with status indicating whether or not the bridge was able to buffer the received DLPDU.

ii) If the destination DL-address specified in the DLPDU is one which the bridge should forward, and the bridge was able to receive and buffer the DLPDU without error, then the received DLPDU shall be forwarded with modification of the frame-control field in the forwarded DLPDU as appropriate (see 6.1.1.3).

iii) If the DLPDU contains an explicit source DL-address, then the bridge shall attempt to update its routing table entry for the source DL-address specified in the DLPDU to reflect the bridge port from which the DLPDU was received.

It is a protocol error for a bridge DLE which will forward the received CA DLPDU to not send an SR reply DLPDU when a reply is required.

NOTE — At most one bridge DLE on the local link should be forwarding the received CA DLPDU.

7.4.4.4 Additional actions required of the current LAS DLE

The LAS DLE shall act as specified in 6.2.7.2.

7.5 COMPEL DATA (CD) DLPDU

A COMPEL DATA (CD) DLPDU is used to request the transfer (or retransfer) of a limited amount of transparent user data from another DLS-user to the requesting DLS-user without requiring that the transaction (request and acknowledgment) occur within the context of a DLC. It is also used for similar purposes within the context of a DLC, and to assist in the synchronization of DLCEPs and of their DLS-users. When the CD DLPDU is addressed to a publishing DLCEP, then the DLS-user-data whose transfer is requested will be distributed to all of the subscribers of the DLC.

A CD DLPDU creates and passes a reply token to the addressed receiving DLE, which upon reception becomes the dominant token on the link. A CD DLPDU requires an immediate reply of either a DATA (DT) DLPDU or a STATUS RESPONSE (SR) DLPDU. If no DLPDU is received in reply, then the transaction is repeated a maximum of V(MRC) times.

7.5.1 Structure of the CD DLPDUs

Table 8 — Structure of CD DLPDUs

format	frame control	destination address	source address	parameters
1L	1111 1FPP	HL.N.S	HL.N.S	o-SD-p
1S	1111 0FPP	N.S	N.S	o-SD-p
2L	1011 1FPP	HL.N.S	—	o-SD-p
2S	1011 0FPP	N.S	—	o-SD-p

7.5.1.1 The frame control field shall specify

a) the DLPDU’s function, including that an immediate reply is being requested;

b) the transaction’s priority and DLPDU’s implied priority;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU, and either its expected immediate reply or the appropriate link-idle timeout, terminates use of a delegated token.

NOTE — This field necessarily has the value NOT-FINAL when an immediate retry of the current transaction is possible. Only a transaction which is guaranteed to not need an immediate retry may have the value FINAL specified in its associated DLPDU.

7.5.1.2 The address field shall consist of either

a) an explicit destination DL-address and an explicit source DL-address, in that order, for formats 1L and 1S; or

b) only an explicit destination DL-address, for formats 2L and 2S.

For formats 1L and 2L, all addresses shall be LONG; for formats 1S and 2S, all addresses shall be SHORT.

7.5.1.3 The status-data-parameters (SD-parameters) field specifies information appropriate to the associated destination DL-address:

a) If that DL-address is a DLSAP-address bound in a responder DL(SAP)-role, then the SD-parameters specify a transaction-id used by the originating DLE to correlate a delayed returned reply with the originating request. This field shall be structured and encoded as described in 8.4.1.

b) If that DL-address is a DLCEP-address, then the SD-parameters specify state information for the addressed DLCEP. This field may be null. When non-null, this field shall be structured and encoded as described in 8.4.2.

NOTE — The size and structure of this field is dependent on the QoS attributes associated with the DLCEP addressed by the destination DL-address specified in this DLPDU, and is determined during DLCEP establishment.

When the CD DLPDU is sent by the LAS DLE, which occurs while the dominant token is a scheduler token, then this field shall be null.

7.5.1.4 The user data field shall be null.

7.5.2 Content of the CD DLPDU

The frame control field shall be encoded as specified in Table 8.

Either the DL-addresses shall be

a) two explicit DLSAP-addresses; or

b) one explicit DLCEP-address, and a second explicit or implicit or not-present DLCEP-address; or

c) one explicit DLCEP-address, followed by one explicit or implicit DLSAP-address.

7.5.2.1 Content of the CD DLPDU when specifying a destination DLSAP-address

When the first DL-address is a DLSAP-address as in 7.5.2(a), then

a) if the DLPDU format is format 1L or 1S, then

1) this DLPDU is being used to implement the unitdata exchange service;

2) the DL(SAP)-role for the destination DLSAP-address shall be CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER;

3) the second address shall be present, shall be a DLSAP-address, and the DL(SAP)-role for that DLSAP-address shall be INITIATOR;

4) the SD-parameters field shall specify a transaction-id used by the originating DLE to correlate a delayed returned reply with the originating request, where the contents of this field shall be as described in 8.4.1; and

5) the user data shall be null.

b) no other DLPDU format may be used.

7.5.2.2 Content of the CD DLPDU when specifying a destination DLCEP-address

When the first address is a DLCEP-address, as in 7.5.2(b) and 7.5.2(c), then

a) this DLPDU shall request state information from the addressed DLCEP, and shall request that DLS-user-data be included in the reply DLPDU;

b) the second address, if present,

— shall be the peer DLCEP-address of the same DLC as the destination peer DLCEP-address; or

— shall be a subscriber DLCEP's calling-DLSAP-address of the same DLC as the destination publisher DLCEP-address; and

c) the SD-parameters field, if present, shall specify state information for the addressed DLCEP, and the contents of this field shall be as described in 8.4.2; and

NOTE 1 — The size and structure of this field is dependent on the QoS attributes associated with the DLCEP addressed by the destination DL-address specified in this DLPDU, and is determined during DLCEP establishment.

d) the user data shall be null.

NOTE 2 — If the user data field is not null, then an ED DLPDU (see 7.6) should be used instead.

NOTE 3 — Formats 1L and 1S are used for peer-to-peer (see 7.5.2(b)) and subscribers-to-publisher (see 7.5.2(c)) communications when the DLPDU-authentication attribute is SOURCE or MAXIMAL. Formats 2L and 2S are used for peer-to-peer and subscribers-to-publisher communications when the DLPDU-authentication attribute is ORDINARY.

7.5.3 Sending the CD DLPDU

A CD DLPDU may be selected for transmission on the link when

a) the sending DLE holds a scheduler token or delegated token which is the dominant token on the local link; and

b) the remaining allocated duration of token usage, C(RD), permits completion of V(MRC)+1 implied transactions prior to expiration of the token, where each transaction consists of sending the CD DLPDU that requires an immediate reply, and awaiting a worst-case SR DLPDU or worst-case permitted DT reply DLPDU containing DLS-user-data; and

c) if the CD DLPDU will be addressed to a DLSAP-address, then the outstanding-transaction-array, V(OTA) (see 5.7.1.15), searched circularly from the last-transaction-index, V(LTI) (see 5.7.1.16), has an unassigned entry whose index is not V(LTI).

Once selected, if the CD DLPDU will be addressed to a DLSAP-address, then

— V(LTI) shall be set to the index of that unassigned entry in V(OTA); and

— that entry in V(OTA) shall be assigned to the selected CD DLPDU and shall record information which permits an expected response DT DLPDU to be correlated with the specific invocation of the unitdata-exchange service which gave rise to the CD DLPDU.

Once selected, transmission of the CD DLPDU shall be retried until either

1) a permissible immediate reply DLPDU is received; or

2) an impermissible DLPDU is received when an immediate reply was expected; or

3) the original transmission and the permitted maximum number of transmission retries, V(MRC) (see 5.7.1.5), have all failed to elicit one of the permissible reply DLPDUs.

If the DLE holds a delegated token, and no additional use of that token after sending this DLPDU and awaiting its immediate reply is needed at that time, then the DLE may set the final-token-use subfield of the CD DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

NOTE – The FINAL transaction is necessarily the (V(MRC)+1)'th in a series; otherwise, if a reply is not received, then another CD DLPDU would have to be sent before the current cycle of token use was completed, making the current transaction NOT-FINAL.

Each explicit DL-address in the CD DLPDU shall be delocalized (see 6.2.2) before transmission.

After sending a CD DLPDU, the sending DLE shall monitor the local link for a reply as specified in 6.2.7.1. The permissible reply DLPDU is either

— a DT DLPDU whose destination DL-address is implicitly or explicitly the originating DLSAP-address or DLCEP specified by the CD DLPDU; or

— a DT DLPDU without a destination DL-address, or

— an SR DLPDU.

If V(LTI) was assigned to the transaction, as a result of the search described in (c), and if a permissible reply DLPDU was not received, then the V(LTI)'th entry shall be deassigned.

7.5.4 Receiving the CD DLPDU

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) upon reception.

A received CD DLPDU shall be treated as follows by the receiving DLE:

7.5.4.1 Actions required of all DLEs

NOTE — The next alternative attempts to detect the reception of a duplicated CD 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 reply DT DLPDU. In such a case the response DT DLPDU is required to be identical to the first in those fields and subfields which convey DLS-user-data and related information.

a) If

1) the destination DL-address specified by the DLPDU designates an active DLSAP-address of the receiving DLE, or an active DLCEP-address of a DLC for which the receiving DLE is peer or publisher;

2) the immediately-prior DLPDU was transmitted as an immediate-response to a received CD DLPDU whose destination DL-address was the same DLSAP-address or DLCEP-address;

3) if a source DL-address is present in the just-received CD DLPDU, then it was also present and identical in that prior received CD DLPDU;

4) starting with the SD-parameters field, the first three octets of the just-received CD DLPDU, or the remainder of the DLPDU, if fewer than three octets, are identical to the corresponding octets of that prior received CD DLPDU; and

5) a period of link inactivity of (immediate-response-recovery-delay + 1) slot-times,

(V(IRRD) + 1) × V(ST) octet-durations, has not occurred since receipt of that prior received CD DLPDU,

then the receiving DLE

i) shall discard the received DLPDU and not forward it to the DLE's upper-level functions for further processing; and

ii) shall retransmit the prior-transmitted immediate-reply DT DLPDU, within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of receipt of the CD DLPDU, unchanged from the prior transmission except for those subfields of

— the final-token-use subfield of the frame control octet which conveys information to the listening LAS; and

— those subfields of any present DLCEP SD-parameters which acknowledge receipt of, or request retransmission of, received DLSDUs.

NOTE — This requirement is meant to ensure that the DL-priority, content and identity of any DLS-user-data conveyed in the immediate reply DT DLPDU is identical to that in the immediately-prior DT DLPDU sent from this same DL-address, including the originator's transaction-id if one is present in the received and transmitted SD-parameters.

b) If (a) does not apply, and the destination DL-address specified by the DLPDU designates a DLSAP-address of the receiving DLE, then the processing of the received DLPDU shall be based upon the DL(SAP)-role specified for that DLSAP-address.

NOTE — Group DL-addresses are not covered here; such CD DLPDUs are erroneous and are unrecognized upon receipt.

The receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of receipt of the CD DLPDU:

1) If the DLE can prepare it in time, then the reply DLPDU shall be a DT DLPDU with any explicit destination DL-address in the DT DLPDU having the same value as the source DL-address of the received CD DLPDU, and length and format as follows:

— DT (see 7.7.1 format 2L) in response to CD (see 7.5.1 format 1L);

— DT (see 7.7.1 format 2S) in response to CD (see 7.5.1 format 1S) if the DT DLPDU contains DLS-user-data; or

— DT (see 7.7.1 format 4) in response to CD (see 7.5.1 format 1S) if the DT DLPDU does not contain DLS-user-data,

and shall include SD-parameters specifying

— the originator's transaction-id as received in the stimulating CD DLPDU, as specified in 7.5.2.1(a.4); and

— status for the DLSDU requested by that CD DLPDU.

The reply DT DLPDU may include a DLSDU which was already buffered at that responding DLE at the time of CD reception.

NOTE — This restriction prohibits the reply DLPDU from reflecting any higher-layer (than DLL) processing of the received DLSDU. This restriction is necessary to enable migration from prior national standards.

If the DL(SAP)-role of the destination DLSAP-address specified

i) BASIC or INITIATOR, then the DLE shall reject the received CD DLPDU based on this DL(SAP)-role, and an appropriate error status without an accompanying DLSDU shall be included in the reply DT DLPDU;

ii) UNCONSTRAINED RESPONDER, then the SD-parameter status shall indicate the highest DLL priority sending buffer with a non-null DLSDU which was available at the addressed DLSAP at the time of reception of the CD DLPDU; and

A) if that DLL priority is greater than or equal to the priority specified in 7.5.1.1(b), then that DLSDU shall be included in the reply DT DLPDU and its buffer set to empty if so configured, and the DLL priority of that reply DT DLPDU shall be the DLL priority of the conveyed DLSDU; or

B) if that DLL priority is less than the priority specified in 7.5.1.1(b), then no DLSDU shall be included in the reply DT DLPDU, and the DLL priority of that reply DT DLPDU shall be the DLL priority specified in 7.5.1.1(b); or

C) if there is such no sending buffer with a non-null DLSDU, then the reply DT DLPDU shall specify an appropriate error status, the DLL priority of that reply DT DLPDU shall be the DLL priority specified in 7.5.1.1(b), and that reply DT DLPDU shall not contain DLS-user-data.

iii) CONSTRAINED RESPONDER, then

A) if the source DLSAP-address of the CD DLPDU is equal to the remote DLSAP address which was specified in the prior DL-BIND request primitive for the receiving DLSAP-address (or its DL-management equivalent), both after delocalization, then the procedure specified in (ii) shall be followed;

B) otherwise, the received DLSDU shall be discarded, and an appropriate error status without an accompanying DLSDU shall be included in the reply DT DLPDU.

2) If the DLE cannot prepare the required reply DT DLPDU in time, then the DLE shall send a DT DLPDU with any explicit destination DL-address in the DT DLPDU having the same value as the source DL-address of the received CD DLPDU, and length and format as follows:

— DT (see 7.7.1 format 2L) in response to CD (see 7.5.1 format 1L); or

— DT (see 7.7.1 format 4) in response to CD (see 7.5.1 format 1S),

and shall include SD-parameters

— with a transaction-id identical to the transaction-id from the received CD DLPDU, as specified in 7.5.2.1(a.4);

— with a status (DR — “delayed reply”) indicating that the DLE requires additional time to prepare the required response; and

— with a null user data field,

and the DLE

— shall prepare that DT DLPDU as specified in (1) as soon as possible; and

— shall include an explicit destination address in the reply DT DLPDU; and

— shall append that reply DT DLPDU to the DLE’s Q(US) to be transmitted at the first opportunity.

c) If (a) does not apply, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE

— is a peer or publisher, and the DLL priority of the DLCEP is not equal to the priority specified in the received DLPDU; or

— is a peer, and the length and number of DL-addresses is not as expected or the DLPDU specifies an explicit source address which is not equal to the remote peer DLCEP’s DLCEP-address,

then

1) the received DLPDU shall be forwarded to the DLE’s upper-level functions (see 9.2.2.9) for further processing; and

2) the receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, $V(MRD) \times V(ST)$ octet-durations, of receipt of the CD DLPDU. The reply DLPDU shall be a DT DLPDU in the format negotiated for the DLC for the selected direction of transmission, and shall contain SD-parameters appropriate to the sending DLCEP, but shall not contain DLS-user-data.

d) If neither (a) nor (c) applies, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE is a peer or publisher, then

1) the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing; and

2) the receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of receipt of the CD DLPDU. The reply DLPDU shall be a DT DLPDU in the format negotiated for the DLC for the selected direction of transmission, shall contain SD-parameters appropriate to the sending DLCEP, and shall contain DLS-user-data if any was available and waiting for transmission or retransmission from the DLCEP; and

NOTE – Migration of prior national standards requires that this data always be included when available.

3) The requesting CD DLPDU may contain information about the state of the requesting DLCEP. The reply DT DLPDU is permitted, but is not required, to reflect that state information in its reply; immediate processing of that state information before sending the immediate reply shall be permitted but shall not be required.

e) If (a) does not apply, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE is a subscriber, then

1) the non-DLS-user-data portion of the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing; and

2) the receiving DLE shall record the destination DL-address from the received CD DLPDU in V(RA) for subsequent association with the expected immediate reply DT DLPDU, which should be the next DLPDU received; and

3) the receiving DLE shall monitor the local link for a reply and then act based on the result of that monitoring, all as specified in 6.2.7.3.

7.5.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.5.4.3 Additional actions required of a Bridge class DLE

If all of the following conditions hold

a) the destination DL-address specified in the DLPDU is one which the bridge should forward but to which the bridge DLE itself would not otherwise generate an immediate reply DLPDU;

b) the immediately-prior DLPDU was transmitted as an immediate-response to a received CD DLPDU with the same destination DL-address;

c) if a source DL-address is present in the just-received CD DLPDU, then it was also present and identical in that prior received CD DLPDU;

d) the basic-DLC-parameters portion of the SD-parameters of the just-received CD DLPDU are identical to those of that prior received CD DLPDU; and

e) a period of link inactivity of immediate-response-recovery-delay plus one slot-times,

(V(IRRD)+1) × V(ST) octet-durations, has not occurred since receipt of that prior received CD DLPDU,

then the bridge

1) shall discard the received DLPDU and not forward it to the bridge's other functions for further processing; and

2) shall initiate retransmission of the prior-transmitted immediate-reply SR DLPDU, unchanged from the prior transmission.

NOTE — This requirement is meant to ensure that any status conveyed in the immediate reply SR DLPDU is identical to that in the immediately-prior SR DLPDU.

Otherwise

i) If the destination DL-address specified in the DLPDU is one which the bridge should forward but which the bridge DLE itself would not otherwise receive, then the bridge shall form and send an SR DLPDU,

— within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of receipt of the CD DLPDU; and

— with status indicating whether or not the bridge was able to buffer the received DLPDU.

ii) If the destination DL-address specified in the DLPDU is one which the bridge should forward, and the bridge was able to receive and buffer the DLPDU without error, then the received DLPDU shall be forwarded with modification of the frame-control field in the forwarded DLPDU as appropriate (see 6.1.1.3).

iii) If the DLPDU contains an explicit source DL-address, then the bridge shall attempt to update its routing table entry for the source DL-address specified in the DLPDU to reflect the bridge port from which the DLPDU was received.

It is a protocol error for a bridge DLE which will forward the received CD DLPDU to not send an SR reply DLPDU when a reply is required.

NOTE — At most one bridge DLE on the local link should be forwarding the received CD DLPDU.

7.5.4.4 Additional actions required of the current LAS DLE

The LAS DLE shall act as specified in 6.2.7.2.

7.6 EXCHANGE DATA (ED) DLPDU

An EXCHANGE DATA (ED) DLPDU is used

a) to transfer (or retransfer) a limited amount of transparent user data from the requesting DLS-user to another DLS-user; and

b) to request the transfer (or retransfer) of a limited amount of transparent user data from that other DLS-user to the requesting DLS-user,

without requiring that the transaction (request and acknowledgment) occur within the context of a DLC. It is also used for similar purposes within the context of a DLC, and to assist in the synchronization of DLCEPs and of their DLS-users. When the ED DLPDU is addressed to a publishing DLCEP, then the DLS-user-data whose transfer is requested in (b) will be distributed to all of the subscribers of the DLC.

An ED DLPDU creates and passes a reply token to the addressed receiving DLE, which upon reception becomes the dominant token on the link. An ED DLPDU requires an immediate reply of either a DATA (DT) DLPDU or a STATUS RESPONSE (SR) DLPDU. If no DLPDU is received in reply, then the transaction is repeated a maximum of V(MRC) times.

7.6.1 Structure of the ED DLPDUs

Table 9 — Structure of ED DLPDUs

format	frame control	destination address	source address	parameters	user data
1L	1100 1FPP	HL.N.S	HL.N.S	SD-p	pDLSDU
1S	1100 0FPP	N.S	N.S	SD-p	pDLSDU
2L	1000 1FPP	HL.N.S	—	SD-p	pDLSDU
2S	1000 0FPP	N.S	—	SD-p	pDLSDU

7.6.1.1 The frame control field shall specify

a) the DLPDU's function, including that an immediate reply is being requested;

b) the transaction's priority, and the DLPDU's implied priority unless explicitly overridden by 7.6.1.3(a.2);

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU, and either its expected immediate reply or the appropriate link-idle timeout, terminates use of a delegated token.

NOTE — This field necessarily has the value NOT-FINAL when an immediate retry of the current transaction is possible. Only a transaction which is guaranteed to not need an immediate retry may have the value FINAL specified in its associated DLPDU.

7.6.1.2 The address field shall consist of

a) an explicit destination DL-address and an explicit source DL-address, in that order, for formats 1L and 1S; or

b) only an explicit destination DL-address only, for formats 2L and 2S.

For formats 1L and 2L, all addresses shall be LONG; for formats 1S and 2S, all addresses shall be SHORT.

7.6.1.3 The status-data-parameters (SD-parameters) field specifies information appropriate to the associated destination DL-address:

a) If that DL-address is a DLSAP-address bound in a responder DL(SAP)-role, then the SD-parameters specify

1) a transaction-id used by the originating DLE to correlate a delayed returned reply with the originating request; and

2) a DLSDU-priority used to convey the actual priority of the accompanying DLSDU to the responding DLE.

This field shall be structured and encoded as described in 8.4.1.

b) If that DL-address is a DLCEP-address, then the SD-parameters specify state information for the addressed DLCEP. This field shall be structured and encoded as described in 8.4.2.

NOTE — The size and structure of this field is dependent on the QoS attributes associated with the DLCEP addressed by the destination DL-address specified in this DLPDU, and is determined during DLCEP establishment.

7.6.1.4 The user data field size and content are limited by the associated destination DL-address:

a) If that DL-address is a DLSAP-address, then the user data field shall consist of DLS-user-data whose maximum size is limited to the smaller of the maximum DLS-user-data sizes permitted for a DLPDU of the priority specified in 7.6.1.1(b) and 7.6.1.3(a.2), and shall not be null.

b) If that DL-address is a DLCEP-address, then the negotiated DLC attributes for the intended direction of transmission shall permit DLS-user-data to be carried in an ED DLPDU (see 8.1(c.6) and 8.1(d.6)), and the user data field shall consist of DLS-user-data whose maximum size is limited to the smaller of

1) the maximum DLS-user-data size permitted for a DLPDU of the priority specified in 7.6.1.1(b); and

2) the maximum DLSDU size negotiated on the DLC for data transmission to that DLCEP,

and shall not be null.

7.6.2 Content of the ED DLPDU

The frame control field shall be encoded as specified in Table 9.

Either the DL-addresses shall be

a) two explicit DLSAP-addresses; or

b) one explicit DLCEP-address, and a second explicit or implicit DLCEP-address; or

c) one explicit DLCEP-address, followed by one explicit or implicit DLSAP-address.

7.6.2.1 Content of the ED DLPDU when specifying a destination DLSAP-address

When the first DL-address is a DLSAP-address as in 7.6.2(a), then

a) if the DLPDU format is format 1L or 1S, then

1) this DLPDU is being used to implement the unitdata exchange service;

2) the DL(SAP)-role for the destination DLSAP-address shall be CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER;

3) the second address shall be present, shall be a DLSAP-address, and the DL(SAP)-role for that DLSAP-address shall be INITIATOR;

4) the SD-parameters field shall specify a DLSDU-priority and a transaction-id used by the originating DLE to correlate a delayed returned reply with the originating request, where the contents of this field shall be as described in 8.4.1, and the DLSDU-priority shall be the priority of the accompanying user data and shall be the same as specified in 7.6.1.3(a.2) and shall be as least as high as the DLPDU priority specified in 7.6.1.1(b); and

5) the user data shall be a single DLSDU whose size is limited to the maximum size for the priority specified in 7.6.1.4(a), and shall not be null.

b) no other DLPDU format may be used.

7.6.2.2 Content of the ED DLPDU when specifying a destination DLCEP-address

When the first address is a DLCEP-address, as in 7.6.2(b) and 7.6.2(c), then

a) this DLPDU can convey a single or partial DLSDU

— from one peer DLCEP to its corresponding peer DLCEP; or

— from a subscriber DLCEP to its corresponding publisher DLCEP,

shall request state information from the addressed DLCEP, and shall request that DLS-user-data be included in the reply DLPDU, and

b) the second address, if present,

- shall be the peer DLCEP-address of the same DLC as the destination peer DLCEP-address; or

- shall be a subscriber DLCEP's calling-DLSAP-address of the same DLC as the destination publisher DLCEP-address; and

c) the SD-parameters field shall specify state information for the addressed DLCEP, and the contents of this field shall be as described in 8.4.2; and

NOTE 1 — The size and structure of this field is dependent on the QoS attributes associated with the DLCEP addressed by the destination DL-address specified in this DLPDU, and is determined during DLCEP establishment.

d) the user data shall specify those octets of a DLSDU consistent with the negotiated DLSDU size and the segmentation information specified in the accompanying SD-parameters, and shall not be null.

NOTE 2 — If the user data field is null, then a CD DLPDU (see 7.5) should be used instead.

NOTE 3 — Formats 1L and 1S are used for peer-to-peer (7.6.2(b)) and subscribers-to-publisher (see 7.6.2(c)) communications when the DLPDU-authentication attribute is SOURCE or MAXIMAL. Formats 2L and 2S are used for peer-to-peer and subscribers-to-publisher communications when the DLPDU-authentication attribute is ORDINARY.

7.6.3 Sending the ED DLPDU

An ED DLPDU may be selected for transmission on the link when

a) the sending DLE holds a scheduler token or delegated token which is the dominant token on the local link;

b) the remaining allocated duration of token usage, C(RD), permits completion of V(MRC)+1 implied transactions prior to expiration of the token, where each transaction consists of sending the ED DLPDU that requires an immediate reply, and awaiting a worst-case SR DLPDU or worst-case permitted DT reply DLPDU not containing DLS-user-data; and

c) if the ED DLPDU will be addressed to a DLSAP-address, then the outstanding-transaction-array, V(OTA) (see 5.7.1.15), searched circularly from the last-transaction-index, V(LTI) (see 5.7.1.16), has an unassigned entry whose index is not V(LTI).

Once selected, if the ED DLPDU will be addressed to a DLSAP-address, then

— V(LTI) shall be set to the index of that unassigned entry in V(OTA); and

— that entry in V(OTA) shall be assigned to the selected ED DLPDU and shall record information which permits an expected response DT DLPDU to be correlated with the specific invocation of the unitdata-exchange service which gave rise to the ED DLPDU.

Once selected, transmission of the ED DLPDU shall be retried until either

1) a permissible immediate reply DLPDU is received; or

2) an impermissible DLPDU is received when an immediate reply was expected; or

3) the original transmission and the permitted maximum number of transmission retries, V(MRC) (see 5.7.1.5), have all failed to elicit one of the permissible reply DLPDUs.

If the DLE holds a delegated token, and no additional use of that token after sending this DLPDU and awaiting its immediate reply is needed at that time, then the DLE may set the final-token-use subfield of the ED DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

NOTE – The FINAL transaction is necessarily the (V(MRC)+1)'th in a series; otherwise, if a reply is not received, then another ED DLPDU would have to be sent before the current cycle of token use was completed, making the current transaction NOT-FINAL.

Each explicit DL-address in the ED DLPDU shall be delocalized (see 6.2.2) before transmission.

After sending an ED DLPDU, the sending DLE shall monitor the local link for a reply as specified in 6.2.7.1. The permissible reply DLPDU is either

— a DT DLPDU whose destination DL-address is implicitly or explicitly the originating DLSAP-address or DLCEP specified by the ED DLPDU; or

— a DT DLPDU without a destination DL-address; or

— an SR DLPDU.

If V(LTI) was assigned to the transaction, as a result of the search described in (c), and if a permissible reply DLPDU was not received, then the V(LTI)'th entry shall be deassigned.

7.6.4 Receiving the ED DLPDU

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) upon reception.

A received ED DLPDU shall be treated as follows by the receiving DLE:

7.6.4.1 Actions required of all DLEs

NOTE — The next alternative attempts to detect the reception of a duplicated ED 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 reply DT DLPDU. In such a case the response DT DLPDU is required to be identical to the first in those fields and subfields which convey DLS-user-data and related information.

a) If

1) the destination DL-address specified by the DLPDU designates an active DLSAP-address of the receiving DLE, or an active DLCEP-address of a DLC for which the receiving DLE is peer or publisher;

2) the immediately-prior DLPDU was transmitted as an immediate-response to a received ED DLPDU whose destination DL-address was the same DLSAP-address or DLCEP-address;

3) a source DL-address is present in the just-received ED DLPDU, then it was also present and identical in that prior received ED DLPDU;

4) starting with the SD-parameters field, the first three octets of the just-received ED DLPDU, or the remainder of the DLPDU, if fewer than three octets, are identical to the corresponding octets of that prior received ED DLPDU; and

5) a period of link inactivity of (immediate-response-recovery-delay + 1) slot-times,

(V(IRRD) + 1) × V(ST) octet-durations, has not occurred since receipt of that prior received ED DLPDU,

then the receiving DLE

i) shall discard the received DLPDU and not forward it to the DLE's upper-level functions for further processing; and

ii) shall retransmit the prior-transmitted immediate-reply DT DLPDU, unchanged from the prior transmission except for

— the final-token-use subfield of the frame control octet which conveys information to the listening LAS; and

— those subfields of any present DLCEP SD-parameters which acknowledge receipt of, or request retransmission of, received DLSDUs.

NOTE — This requirement is meant to ensure that the DL-priority, content and identity of any DLS-user-data conveyed in the immediate reply DT DLPDU, is identical to that in the immediately-prior DT DLPDU sent from this same DL-address, including the originator's transaction-id if one is present in the received and transmitted SD-parameters.

b) If (a) does not apply, and the destination DL-address specified by the DLPDU designates a DLSAP-address of the receiving DLE, then the processing of the received DLPDU shall be based upon the DL(SAP)-role specified for that DLSAP-address.

NOTE 1 — Group DL-addresses are not covered here; such ED DLPDUs are erroneous and are unrecognized upon receipt.

The receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of receipt of the ED DLPDU:

1) If the DLE can prepare it in time, then the reply DLPDU shall be a DT DLPDU

— with any explicit destination DL-address in the DT DLPDU having the same value as the source DL-address of the received ED DLPDU, and length and format as follows:

- DT (see 7.7.1 format 2L) in response to ED (see 7.6.1 format 1L); or
- DT (see 7.7.1 format 2S) in response to ED (see 7.6.1 format 1S) if the DT DLPDU contains DLS-user-data; or
- DT (see 7.7.1 format 4) in response to ED (see 7.6.1 format 1S) if the DT DLPDU does not contain DLS-user-data,

and shall include SD-parameters specifying

— the originator's transaction-id as received in the stimulating ED DLPDU, as specified in 7.6.2.1(a.4); and

— reception error status for that ED-conveyed DLSDU if the received DLSDU could not be buffered or queued; otherwise status for the DLSDU requested by that ED DLPDU.

The received DLSDU shall be forwarded to the DLE's upper-level functions (see 9.3) for further processing.

If the received DLSDU was buffered or queued successfully, then the reply DLPDU may include a DLSDU which was already buffered at that responding DLE at the time of ED reception.

NOTE 2 — This restriction prohibits the reply DLPDU from reflecting any higher-layer (than DLL) processing of the received DLSDU. This restriction is necessary to enable migration from prior national standards.

If the DL(SAP)-role of the destination DLSAP-address specified

i) BASIC or INITIATOR, then the DLE shall reject the received ED DLPDU based on this DL(SAP)-role, the received DLSDU shall be discarded, and an appropriate error status without an accompanying DLSDU shall be included in the reply DT DLPDU;

ii) UNCONSTRAINED RESPONDER, then

A) if no receiving buffer or queue is explicitly bound to the destination DLSAP-address at the DLL priority of the received DLSDU, or if a receiving queue is explicitly bound but is full, then the received DLSDU shall be discarded, and an appropriate error status without an accompanying DLSDU shall be included in the reply DT DLPDU; or

B) otherwise the received DLSDU shall be put into that buffer or appended to that not-full queue and the SD-parameter status shall indicate the highest DLL priority sending buffer with a non-null DLSDU which was available at the addressed DLSAP at the time of reception of the ED DLPDU, and

I)if that DLL priority is greater than or equal to the priority specified in 7.6.1.1(b), then that DLSDU shall be included in the reply DT DLPDU and its buffer set to empty if so configured, and the DLL priority of that reply DT DLPDU shall be the DLL priority of the conveyed DLSDU;

II)if that DLL priority is less than the priority specified in 7.6.1.1(b), then no DLSDU shall be included in the reply DT DLPDU, and the DLL priority of that reply DT DLPDU shall be the DLL priority specified in 7.6.1.1(b); or

III)if there is no such sending buffer with a non-null DLSDU, then the reply DT DLPDU shall specify an appropriate error status, the DLL priority of that reply DT DLPDU shall be the DLL priority specified in 7.6.1.1(b), and that reply DT DLPDU shall not contain DLS-user-data.

iii) CONSTRAINED RESPONDER, then

A) if the source DLSAP-address of the ED DLPDU is equal to the remote DLSAP address which was specified in the prior DL-BIND request primitive for the receiving DLSAP-address (or its DL-management equivalent), both after delocalization, then the procedure specified in (ii) shall be followed; or

B) otherwise, the received DLSDU shall be discarded, and an appropriate error status without an accompanying DLSDU shall be included in the reply DT DLPDU.

2) If the DLE cannot prepare the required reply DT DLPDU in time, then the DLE shall send a DT DLPDU with any explicit destination DL-address in the DT DLPDU having the same value as the source DL-address of the received ED DLPDU, and length and format as follows:

— DT (see 7.7.1 format 2L) in response to ED (see 7.5.1 format 1L); or

— DT (see 7.7.1 format 4) in response to ED (see 7.5.1 format 1S),

and shall include SD-parameters

— with a transaction-id identical to the transaction-id from the received ED DLPDU, as specified in 7.6.2.1(a.4);

— with a status (DR — "delayed reply") indicating that the DLE requires additional time to prepare the required response; and

— with a null user data field,

and the DLE

— shall prepare that required DT DLPDU as soon as possible, as just described;

— shall include an explicit destination address in the reply DLPDU; and

— shall append that reply DLPDU to the DLE's Q(US) to be transmitted at the first opportunity.

c) If (a) does not apply, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE

— is a peer or publisher, and the DLL priority of the DLCEP is not equal to the priority specified in the received DLPDU; or

— is a peer, and the length and number of DL-addresses is not as expected or the DLPDU specifies an explicit source address which is not equal to the remote peer DLCEP's DLCEP-address,

then

1) the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing; and

2) the receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of receipt of the CD DLPDU. The reply DLPDU shall be a DT DLPDU in the format negotiated for the DLC for the selected direction of transmission, and shall contain SD-parameters appropriate to the sending DLCEP, but shall not contain DLS-user-data.

d) If neither (a) nor (b) applies, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE is a peer or publisher, then

1) the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing;

2) the receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of receipt of the ED DLPDU. The reply DLPDU shall be a DT DLPDU in the format negotiated for the DLC for the selected direction of transmission, shall contain SD-parameters appropriate to the sending DLCEP, and shall contain DLS-user-data, if any

— was available and waiting for transmission or retransmission from the DLCEP; and

— can be included in the DT DLPDU while still meeting the required maximum reply delay; and

3) the requesting ED DLPDU may contain information about the state of the requesting DLCEP. The reply DT DLPDU is permitted, but is not required, to reflect that state information in its reply; immediate processing of that state information before sending the immediate reply shall be permitted but shall not be required.

The received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing.

e) If (a) does not apply, and the destination DL-address specified by the DLPDU designates an active DLCEP-address of a DLC for which the receiving DLE is a subscriber, then

1) the non-DLS-user-data portion of the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2.2.9) for further processing;

2) the receiving DLE shall record the destination DL-address from the received ED DLPDU in V(RA) for subsequent association with the expected immediate reply DT DLPDU, which should be the next DLPDU received; and

3) the receiving DLE shall monitor the local link for a reply and then act based on the result of that monitoring, all as specified in 6.2.7.3.

7.6.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.6.4.3 Additional actions required of a Bridge class DLE

If all of the following conditions hold

a) the destination DL-address specified in the DLPDU is one which the bridge should forward but to which the bridge DLE itself would not otherwise generate an immediate reply DLPDU;

b) the immediately-prior DLPDU was transmitted as an immediate-response to a received ED DLPDU with the same destination DL-address;

c) if a source DL-address is present in the just-received ED DLPDU, then it was also present and identical in that prior received ED DLPDU;

d) the basic-DLC-parameters portion of the SD-parameters of the just-received ED DLPDU are not null, and are identical to those of that prior received ED DLPDU; and

e) a period of link inactivity of immediate-response-recovery-delay plus one slot-times,

$(V(IRRD)+1) \times V(ST)$ octet-durations, has not occurred since receipt of that prior received ED DLPDU,

then the bridge

1) shall discard the received DLPDU and not forward it to the bridge's other functions for further processing; and

2) shall initiate retransmission of the prior-transmitted immediate-reply SR DLPDU, unchanged from the prior transmission.

NOTE — This requirement is meant to ensure that any status conveyed in the immediate reply SR DLPDU is identical to that in the immediately-prior SR DLPDU.

Otherwise,

i) If the destination DL-address specified in the DLPDU is one which the bridge should forward but which the bridge DLE itself would not otherwise receive, then the bridge shall form and send an SR DLPDU

— within a period of maximum-response-delay slot-times, $V(MRD) \times V(ST)$ octet-durations, of receipt of the ED DLPDU; and

— with status indicating whether or not the bridge was able to buffer the received DLPDU.

ii) If the destination DL-address specified in the DLPDU is one which the bridge should forward, and the bridge was able to receive and buffer the DLPDU without error, then the received DLPDU shall be forwarded with modification of the frame-control field in the forwarded DLPDU as appropriate (see 6.1.1.3).

iii) If the DLPDU contains an explicit source DL-address, then the bridge shall attempt to update its routing table entry for the source DL-address specified in the DLPDU to reflect the bridge port from which the DLPDU was received.

It is a protocol error for a bridge DLE which will forward the received ED DLPDU to not send an SR reply DLPDU when a reply is required.

NOTE — At most one bridge DLE on the local link should be forwarding the received ED DLPDU.

7.6.4.4 Additional actions required of the current LAS DLE

The LAS DLE shall act as specified in 6.2.7.2.

7.7 DATA (DT) DLPDU

A DATA (DT) DLPDU is used to transfer a limited amount of transparent user data from one DLS-user to one or more other DLS-users; to acknowledge the transfer of such data; and to assist in the synchronization of DLCEPs and of DLS-users.

It is also used by a responding DLE when replying to a received CA, CD or ED DLPDU, when the time permitted by V(ST) and V(MRD) (see 5.7.1.1 and 5.7.1.2) is inadequate for that DLE to generate the required response to the received DLPDU.

It is also used by a DLE to send an SPDU to one or more other DLEs.

7.7.1 Structure of the DT DLPDUs

Table 10 — Structure of DT DLPDUs

format	**frame control**	**destination address**	**source address**	**parameters**	**user data**
1L	1101 1FPP	HL.N.S	HL.N.S	SD-p	o-pDLSDU
1S	1101 0FPP	N.S	N.S	SD-p	o-pDLSDU
2L	1001 1FPP	HL.N.S	—	SD-p	o-pDLSDU
2S	1001 0FPP	N.S	—	SD-p	o-pDLSDU
3L	0101 1FPP		HL.N.S	SD-p	o-pDLSDU
3S	0101 0FPP		N.S	SD-p	o-pDLSDU
4	1001 0F00	[PSA]	—	SD-p	o-pDLSDU
5	0101 0F00		[PDA]	SD-p	o-pDLSDU

LEGEND:

[PDA] is the destination DL-address from the immediately-prior CA, CD or ED DLPDU

[PSA] is the implied source DL-address from the immediately-prior CA, CD or ED DLPDU

7.7.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's priority;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU terminates use of a delegated token.

7.7.1.2 The address field shall consist of either

a) an explicit destination DL-address and an explicit source DL-address, in that order, for formats 1L and 1S;

b) only an explicit destination DL-address, for formats 2L and 2S;

c) only an explicit source DL-address, for formats 3L and 3S;

d) only an implied destination DL-address, for format 4; or

e) only an implied source DL-address, for format 5.

For formats 1L, 2L and 3L, all addresses shall be LONG; for formats 1S, 2S and 3S, all addresses shall be SHORT; for formats 4 and 5 all addresses shall be VERY-SHORT (see 6.2.1.1) (that is, implicit).

7.7.1.3 When the destination DL-address explicitly specifies a group DL-address, the status-data-parameters (SD-parameters) field is null (formats 1L and 1S), as described in 8.4.1.

When the destination DL-address explicitly or implicitly specifies a DLSAP-address, the SD-parameters field specifies a transaction-id and status (formats 2L, 2S, 4), or is null (formats 1L and 1S), as described in 8.4.1.

When the first DL-address explicitly or implicitly specifies a DLCEP-address, the SD-parameters field specifies state information for the DLCEP addressed by the destination DL-address (formats 1L, 1S, 2L, 2S, 4), or source DL-address (formats 3L, 3S, 5), as described in 8.4.2. The size and structure of this field is dependent on the QoS attributes associated with the addressed DLCEP, and is determined during DLCEP establishment.

7.7.1.4 The user data field shall consist of a single or partial optional DLSDU whose maximum size is limited to the smaller of

a) the maximum DLS-user-data size permitted for a DLPDU of the priority specified in 7.7.1.1(b); and

b) when the DLPDU's explicit or implied destination (or source) address is a DLCEP-address, the maximum DLSDU size negotiated on the DLC for this direction of data transmission to (or from) that DLCEP.

7.7.2 Content of the DT DLPDU

The frame control field shall be encoded as specified in Table 10.

Either the DL-addresses shall be

a) the first a group DL-address and the second a DLSAP-address;

b) all DLSAP-addresses;

c) all DLCEP-addresses; or

d) the first a DLCEP-address and the second a DLSAP-address.

7.7.2.1 Content of the DT DLPDU when specifying a destination DL(SAP)-address

When the first address is a group DL-address as in 7.7.2(a), then

a) If the DLPDU format is format 1L or 1S, then

   1) the DLPDU is being used to implement the unitdata transfer service;

   2) the DL(SAP)-role for the destination DLSAP-address shall be GROUP;

   3) the DL(SAP)-role for the source DLSAP-address shall be BASIC;

   4) the SD-parameters field shall be null; and

5) the user data shall be a single DLSDU whose size is limited to the maximum size for the priority specified in 7.7.1.1(b), and shall not be null.

b) No other DLPDU format may be used.

When the addresses are DLSAP-addresses as in 7.7.2(b), then

c) If the DLPDU format is format 1L or 1S, then

1) this DLPDU is being used to implement the unitdata transfer service;

2) the DL(SAP)-role for the destination DLSAP-address shall be BASIC;

3) the DL(SAP)-role for the source DLSAP-address shall be BASIC;

4) the SD-parameters field shall be null; and

5) the user data shall be a single DLSDU whose size is limited to the maximum size for the priority specified in 7.7.1.1(b), and shall not be null.

d) If the DLPDU format is format 2L, 2S or 4, then

1) this DLPDU conveys an acknowledgment or reply as part of the unitdata transfer (with remote-DLE-confirmation) service, or the unitdata exchange service;

2) the DL(SAP)-role for the destination DLSAP-address shall be BASIC or INITIATOR;

3) the DL(SAP)-role for the implicit source DLSAP-address is BASIC or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER;

4) the SD-parameters field shall specify a transaction-id and reply status for the addressed DLSAP, and the contents of this field shall be as described in 8.4.1; and

5) the user data either shall be a single DLSDU whose size is limited to the maximum size for the priority specified in 7.7.1.1(b), or shall be null, as follows:

— formats 2L may be null or non-null;

— format 2S shall always be non-null; and

— format 4 shall always be null.

e) No other DLPDU format may be used.

7.7.2.2 Content of the DT DLPDU when specifying a destination or source DLCEP-address

When the first address is a DLCEP-address, as in 7.7.2(c) and 7.7.2(d), then

a) this DLPDU can convey a single or partial DLSDU

— from one peer DLCEP to its corresponding peer DLCEP;

— from a subscriber DLCEP to its corresponding publisher DLCEP; or

— from a publisher DLCEP to its corresponding subscriber DLCEPs,

and

b) the second address, if present,

— shall be the peer DLCEP-address of the same DLC as the destination DLCEP-address; or

— shall be a subscriber DLCEP's calling-DLSAP-address; and

c) the SD-parameters field shall specify state information for the addressed DLCEP, and the contents of this field shall be as described in 8.4.2;

NOTE — The size and structure of this field is dependent on the QoS attributes associated with the DLCEP addressed by the destination DL-address specified in this DLPDU, and is determined during DLCEP establishment.

d) the user data shall specify those octets of a DLSDU consistent with the negotiated DLSDU size and the segmentation information specified in the accompanying SD-parameters, and may be null.

NOTES —

1. Formats 1L, 2L, 1S and 2S are used for peer-to-peer and subscribers-to-publisher communications; 1L is used when the DLPDU-authentication attribute is SOURCE or MAXIMAL; 1S is used when the DLPDU-authentication attribute is SOURCE; 2L and 2S are used when the DLPDU-authentication attribute is ORDINARY.

Format 3L is used for publisher-to-subscriber communications when the DLPDU-authentication attribute is MAXIMAL. Formats 3L and 3S are used for publisher-to-subscriber communications when the DLPDU-authentication attribute is ORDINARY or SOURCE.

The specific format to be used (of formats 1L to 3S) is determined as part of DLCEP establishment.

2. Formats 4 and 5 can be used instead of formats 2S and 3S, respectively, only when the sending DLE holds a reply token and when the DLPDU-authentication attribute is ORDINARY.

7.7.3 Sending the DT DLPDU

A DT DLPDU may be selected for transmission on the link when the sending DLE

a) has just received a reply token in a CA, CD or ED DLPDU, permitting a single transmission of a DT or SR DLPDU; or

b) holds a scheduler token or delegated token which is the dominant token on the local link, and when the remaining allocated duration of token usage, C(RD), permits completion of the DT DLPDU's transmission prior to expiration of the token.

Each explicit DL-address in the DLPDU shall be delocalized (see 6.2.2) before transmission.

7.7.3.1 Transmission when the reply token is dominant

a) A DT DLPDU may be sent on the link when the sending DLE has received a CD or ED DLPDU addressed

— to one of its active DLSAP-addresses; or

— to one of its active DLCEP-addresses for which it has a peer or publisher DLCEP,

and the sending DLE is replying as specified by 7.5.4.1 or 7.6.4.1, by forming as an immediate reply a DT DLPDU which may include a DLSDU which was already buffered or queued at that responding DLE at the time of the CD or ED DLPDU's reception.

NOTE — This restriction prohibits the reply DLPDU from reflecting any higher-layer processing of the received DLSDU. This restriction is necessary to enable migration from prior national standards.

b) A DT DLPDU may be sent on the link when the sending DLE has received a CA DLPDU addressed

— to one of its active DLSAP-addresses; or

— to one of its active DLCEP-addresses for which it has a peer or publisher DLCEP,

and the sending DLE is replying as specified by 7.4.4.1, by forming as an immediate reply a DT DLPDU which does not include a DLSDU.

When an immediate reply to a CA, CD or ED DLPDU is required, as specified in (a) or (b), then the replying DLE shall send a reply DT DLPDU within a period of maximum-response-delay slot-times, V(MRD) $\times$ V(ST) octet-durations, of receipt of the requesting CA, CD or ED DLPDU.

If the CA, CD or ED DLPDU which required the reply was addressed to a DLCEP, then that CA, CD or ED DLPDU may have contained SD-parameters which conveyed information about the state of the sending DLCEP to the receiving DLCEP. The receiving DLE is permitted, but is not required, to reflect that state information in its reply DT DLPDU; immediate processing of that state information before sending the immediate reply shall be permitted but shall not be required.

The final-token-use subfield of the reply DT DLPDU shall have the same value as that in the requesting CA, CD or ED DLPDU.

Each explicit DL-address in the reply DT DLPDU shall be delocalized (see 6.2.2) before transmission.

It is a protocol error for the addressed DLE to not send a DT reply DLPDU when a reply is required.

NOTE — At most one DLE on the local link should be sending a reply to the received CA, CD or ED DLPDU. That reply may be either a DT or SR DLPDU.

7.7.3.2 Transmission when the delegated token is dominant

A DT DLPDU may be sent on the link when the sending DLE holds a delegated token which is the dominant token on the local link, and when the remaining allocated duration of token usage, C(RD), permits completion of the DLPDU’s transmission prior to expiration of the token.

If no additional use of that token after sending this DLPDU is needed at that time, then the DLE may set the final-token-use subfield of the DT DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

Each explicit DL-address in the DLPDU shall be delocalized (see 6.2.2) before transmission.

7.7.3.3 Transmission when the scheduler token is dominant

The LAS DLE may send a DT DLPDU on the link when both the remaining allocated duration of token usage before the next scheduled activity, and the remaining allocated duration of scheduler token usage for link maintenance, which is V(LTHT) octet-durations minus the amount of link capacity used for link maintenance during the current cycle of “circulating the token,” permits completion of the DLPDU’s transmission prior to expiration of the token.

Each explicit DL-address in the DLPDU shall be delocalized (see 6.2.2) before transmission.

7.7.4 Receiving the DT DLPDU

Each DL-address in the DLPDU shall be delocalized (see 6.2.2) upon reception.

A received DT DLPDU shall be treated as follows by the receiving DLE:

7.7.4.1 Actions required of all DLEs

7.7.4.1.1 Actions required when the reply token was not dominant at start-of-reception

a) If the received DT DLPDU has format 1L or 1S, and its destination DL-address designates a DL(SAP)-address of the receiving DLE, then the received DLPDU shall be forwarded to the DLE’s upper-level functions (see 9.3.1.3) for further processing.

b) If the received DT DLPDU has format 2L or 2S, and its destination DL-address designates a DLSAP-address of the receiving DLE, and the transaction-id reported by the SD-parameters indexes an assigned transaction in the outstanding-transaction-array, V(OTA) (see 5.7.1.15), then

— the DLE shall consider the prior transmission to have been error-free;

— the information in the indexed entry in V(OTA) shall be associated with the received DT DLPDU;

— the received DT DLPDU and that associated information shall be forwarded to the DLE's upper-level functions (see 9.3) for further processing; and

— the indexed entry in V(OTA) shall be unassigned.

c) If the received DT DLPDU

1) has format 1L, 1S, 2L or 2S, and its destination DL-address designates a DLCEP-address designating a peer or publisher DLCEP of the receiving DLE; or

2) has format 3L or 3S, and its source DL-address designates a DLCEP-address designating a subscriber DLCEP of the receiving DLE,

then the received DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2) for further processing.

d) If the received DT DLPDU has format 1L or 1S, and its destination DL-address designates NODE.0 DL-address, V(TN).0, of the receiving DLE's DL-support functions, then the received DLPDU shall be forwarded to the DLE's upper-level functions (see 10.3) for further processing.

e) If none of (a), (b), (c) or (d) applies, then the DT DLPDU shall be reported to local DL-management as an unexpected response, and shall be discarded.

NOTE — This report may take the form of incrementing a DL-management error counter.

7.7.4.1.2 Actions required when the reply token was dominant at start-of-reception and the receiving DLE sent the CA, CD or ED DLPDU which created the reply token

a) If the received DT DLPDU

— has format 2L or 2S, and its destination DL-address designates a DLSAP-address of the receiving DLE; or

— has format 4, and the source DL-address from the immediately-prior CA, CD or ED DLPDU was a DLSAP-address,

then

1) If the status reported by the SD-parameters is "DR — delayed reply" (that is, has a value of F_{16}), then the DLE

— shall consider the prior transmission to have been error-free; and

— shall discard the DLPDU.

NOTE — The DLE should expect to receive another DT DLPDU at a later time, with the same destination DL-address and format, as a delayed reply.

2) If (1) does not apply, then

— the DLE shall consider the prior transmission to have been error-free;

— the information in the V(LTI)'th entry in V(OTA) (see 5.7.1.16 and 5.7.1.15) shall be associated with the received DT DLPDU;

— the received DT DLPDU and that associated information shall be forwarded to the DLE's upper-level functions (see 9.3) for further processing; and

— the V(LTI)'th entry in V(OTA) shall be deassigned.

b) If the received DT DLPDU has format 1L, 1S, 2L or 2S, and its destination DL-address designates a DLCEP-address of a peer DLCEP of the receiving DLE, then

— the DLE shall consider the prior transmission to have been error-free; and

— the received DT DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2) for further processing.

c) If the received DT DLPDU has format 4, and the explicit or implicit source DL-address from the immediately-prior CA, CD or ED DLPDU was a DLCEP-address of a peer DLCEP of the receiving DLE, then

— the DLE shall consider the prior transmission to have been error-free; and

— the received DT DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2) for further processing, with its implied destination DLCEP-address assumed to be the explicit or implicit source DL-address from that immediately-prior CA, CD or ED DLPDU.

d) If the received DT DLPDU has format 3L or 3S, and its source DL-address designates the publisher's DLCEP-address of a subscriber DLCEP of the receiving DLE, and this source DL-address is equal to the destination DL-address from the immediately-prior CA, CD or ED DLPDU, then

— the DLE shall consider the prior transmission to have been error-free; and

— the received DT DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2) for further processing.

e) If the received DT DLPDU has format 5, and the explicit destination DL-address from the immediately-prior CA, CD or ED DLPDU was a DLCEP-address of a subscriber DLCEP of the receiving DLE, then

— the DLE shall consider the prior transmission to have been error-free; and

— the received DT DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2) for further processing, with its implied source DLCEP-address assumed to be the explicit destination DL-address from that immediately-prior CA, CD or ED DLPDU.

f) If none of (a) through (e) applies, then the DT DLPDU shall be reported to local DL-management as an unexpected response, and shall be discarded.

NOTE — This report may take the form of incrementing a DL-management error counter.

7.7.4.1.3 Actions required when the reply token was dominant at start-of-reception and the receiving DLE did not send the CA, CD or ED DLPDU which created the reply token

a) If the received DT DLPDU has format 3L or 3S, and its source DL-address designates a publisher's DLCEP-address of a subscriber DLCEP of the receiving DLE, and this source DL-address is equal to the destination DL-address, V(RA), from the immediately-prior CA, CD or ED DLPDU, then

— the DLE shall consider the prior transmission to have been error-free; and

— the received DT DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2) for further processing.

b) If the received DT DLPDU has format 5, and the explicit destination DL-address from the immediately-prior CA, CD or ED DLPDU was the publisher's DLCEP-address of a subscriber DLCEP of the receiving DLE, then

— the DLE shall consider the prior transmission to have been error-free; and

— the received DT DLPDU shall be forwarded to the DLE's upper-level functions (see 9.2) for further processing, with its implied source DLCEP-address assumed to be the explicit destination DL-address from that immediately-prior CA, CD or ED DLPDU.

7.7.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.7.4.3 Additional actions required of a Bridge class DLE

a) If the first DL-address specified in the DLPDU is an explicit DL-address to which the bridge should forward the DLSDU and the bridge was able to buffer the DLPDU without error, then the DLPDU shall be forwarded with modification of the frame-control field in the forwarded DLPDU as appropriate (see 6.1.1.3).

b) If the DLPDU contains an explicit source DL-address, then the bridge shall attempt to update its routing table entry for the source DL-address specified in the DLPDU to reflect the bridge port from which the DLPDU was received.

c) Otherwise, if neither (a) nor (b) applies, then the DLE shall not forward the DLPDU.

7.7.4.3.1 Actions required when the reply token was dominant at start-of-reception and the receiving bridge DLE forwarded, but did not originate, the CA, CD or ED DLPDU which created the reply token

a) The DLE shall consider the prior transmission to have been error-free.

b) If

— the received DT DLPDU has format 4;

— the destination DL-address from the immediately-prior CA, CD or ED DLPDU was a DLSAP-address;

— the status reported by the SD-parameters is other than "DR — delayed reply" (that is, has a value other than F_{16}); and

— the user-data field is null,

then the DLE shall form a DT DLPDU

1) with format 2L when the previous CA, CD or ED DLPDU was format 1L, or format 2S when the previous CA, CD or ED DLPDU was format 1S;

2) with the explicit destination DL-address equal to the source DL-address in that immediately prior CA, CD or ED DLPDU;

3) with an SD-parameters field equal to the SD-parameters field of the received DT DLPDU; and

4) with a null user-data field;

and shall forward the just-formed DT DLPDU as if it had just been originated by the upper-level functions of the bridge DLE.

NOTE — This means that the bridge DLE originates the FCS for the newly-formed DT DLPDU as in 6.1.1.1, rather than just modifying a previously-received FCS during normal forwarding as in 6.1.1.3.

7.7.4.4 Additional actions required of the current LAS DLE

If the final-token-use subfield of the received DLPDU has the value FINAL, then the LAS DLE shall

a) assume that the current use of the delegated token is terminated as if the token had been returned by an RT (see 7.17) DLPDU; and

b) assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.8 STATUS RESPONSE (SR) DLPDU

A STATUS RESPONSE (SR) DLPDU is sent only while holding a reply token; it is used

a) to indicate the receipt of the immediately prior CA, CD or ED DLPDU by the bridge which would normally forward that DLPDU toward the addressed DLE, to indicate to the sending DLE that no error occurred, or that the indicated error occurred; and

b) to reject an attempted transfer of the LAS role from the current LAS DLE to another link-master DLE.

7.8.1 Structure of the SR DLPDU

Table 11 — Structure of SR DLPDUs

frame control	**destination address**	**source NODE-address**	**parameters**
0001 0F11	[PSA]	N	o-SR-p

LEGEND:

[PSA] is the implied DL-address equal to the implied or explicit source DL-address of the immediately prior DLPDU on the link

7.8.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is that of the immediately-prior DLPDU on the local link (to which the SR DLPDU is an immediate response);

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU terminates use of a delegated token.

7.8.1.2 The address field shall consist of an explicit source NODE DL-address.

7.8.1.3 The delayed-reply-parameters (SR-parameters) field specifies an optional transaction-id and status. When non-null, this field shall be structured and encoded as described in 8.5.

7.8.1.4 The user-data field shall be null.

7.8.2 Content of the SR DLPDU

The frame control field shall be encoded as specified in Table 11:

— When sent as an immediate reply to a TL DLPDU, the final-token-use subfield of the SR DLPDU shall specify FINAL.

— When sent as an immediate reply to a CA, CD or ED DLPDU, the final-token-use subfield of the SR DLPDU shall have the same value as that in the requesting CA, CD or ED DLPDU.

The address field shall be the replying DLE's NODE DL-address, whose value is the sender's node-id, V(TN).

NOTE — The primary purposes for the inclusion of the NODE DL-address is to identify the DLE which has assumed the reply token to any observing link analyzer.

When sent as an immediate reply to a CA, CD or ED DLPDU,

a) the SR-parameters shall be null when the DLE is a bridge DLE which has been able to accept the transaction-initiating DLPDU for forwarding;

b) otherwise, when (a) does not apply, the SR-parameters shall be encoded as described in 8.5.

When sent as an immediate reply to a TL DLPDU, the SR-parameters shall be encoded as described in 8.5.

7.8.3 Sending the SR DLPDU

a) An SR DLPDU may be sent on the link when the sending DLE is a bridge DLE which has just received a reply token in a CA, CD or ED DLPDU, permitting that DLE to transmit a single DT or SR DLPDU, and the DLE is replying as specified by 7.4.4.3, 7.5.4.3 or 7.6.4.3.

It is a protocol error for a bridge DLE which will forward the received CA, CD or ED DLPDU to not send an SR reply DLPDU when a reply is required.

NOTE — At most one bridge DLE on the local link should be forwarding the received CA, CD or ED DLPDU.

b) An SR DLPDU may be sent on the link when the sending DLE is a link master or bridge DLE which has just received a reply token in a TL DLPDU, and the receiving DLE needs to reject the transfer of the LAS role (see 7.20.4.2).

The final-token-use subfield of the reply SR DLPDU shall have the same value as that in the requesting CA, CD or ED DLPDU, or shall have the value FINAL for a reply to a requesting TL DLPDU.

When an immediate reply to a CA, CD, ED or TL DLPDU is required, then the DLE shall reply within a period of maximum-response-delay slot-times, V(MRD) $\times$ V(ST) octet-durations, of receipt of the requesting CA, CD, ED or TL DLPDU.

7.8.4 Receiving the SR DLPDU

7.8.4.1 Actions required of all DLEs

a) A received SR DLPDU, received as a reply to an immediately prior CA, CD or ED DLPDU which was originated by the receiving DLE and which was addressed to a DLSAP-address, shall cause the receiving DLE

1) to consider the prior transmission to have been error-free; and

2) to forward that received DLPDU to the DLE's upper-level functions (see 9.3 for further processing).

b) A received SR DLPDU, received as a reply to an immediately prior TL DLPDU which was originated by the receiving (LAS) DLE shall cause the receiving DLE

1) to consider the prior transmission to have been error-free; and

2) as specified in 7.20.3

— to re-assume the scheduler token;

— to inform local DL-management of the event; and

— to resume active operation as the LAS, and commence transmission on the link.

7.8.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.8.4.3 Additional actions required of a Bridge class DLE

a) A received SR DLPDU

1) which reports a status other than “BOK — bridge OK” (that is, has a value other than E_{16}); and

2) which was received as a reply to an immediately prior CA, CD or ED DLPDU which was forwarded (and therefore not originated) by the receiving DLE and which was addressed to a DLSAP-address

shall cause the receiving DLE to form a DT DLPDU

i) with format 2L when the previous CA, CD or ED DLPDU was format 1L, or format 2S when the previous CA, CD or ED DLPDU was format 1S;

ii) with the explicit destination DL-address equal to the source DL-address in that immediately prior CA, CD or ED DLPDU;

iii) with an SD-parameters field equal to the SR-parameters field of the received SR DLPDU; and

iv) with a null user-data field;

and shall forward the just-formed DT DLPDU as if it had just been originated by the upper-level functions of the bridge DLE.

NOTE — This means that the bridge DLE originates the FCS for the newly-formed DT DLPDU as in 6.1.1.1, rather than just modifying a previously-received FCS during normal forwarding as in 6.1.1.3.

b) Otherwise, if (a) does not apply, then the DLE shall not forward the DLPDU.

7.8.4.4 Additional actions required of the current LAS DLE

a) If the final-token-use subfield of the received DLPDU has the value FINAL, then the LAS DLE shall

1) assume that the current use of the delegated token is terminated as if the token had been returned by an RT (see 7.17) DLPDU; and

2) assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

b) A received SR DLPDU, received as a reply to an immediately prior TL DLPDU which was originated by the LAS DLE, shall cause the LAS DLE

1) to consider the prior transmission to have been error-free;

2) to terminate the attempted transfer of the LAS role; and

3) to inform DL-management of the event.

7.9 COMPEL TIME (CT) DLPDU

A COMPEL TIME (CT) DLPDU is used by the current holder of a delegated token to request the LAS DLE to transmit as an immediate response a TIME DISTRIBUTION (TD) DLPDU, thus enabling all other DLEs on the local link

— to update their sense of DL-time; and

— to synchronize the rates of advance of their senses of time.

7.9.1 Structure of the CT DLPDU

Table 12 — Structure of CT DLPDUs

frame control
0001 0F00

7.9.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is URGENT;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU, and either its expected immediate reply or the appropriate link-idle timeout, terminates use of a delegated token.

7.9.1.2 The address field shall be null.

7.9.1.3 The parameters field shall be null.

7.9.1.4 The user-data field shall be null.

7.9.2 Content of the CT DLPDU

The frame control field shall be encoded as specified in Table 12.

7.9.3 Sending the CT DLPDU

A DLE's need for the current DL-time, or to synchronize with the LAS DLE and that DLE's rate of advance of the current DL-time, may result in transmission of a CT DLPDU. A CT DLPDU requests that the LAS DLE send a TD DLPDU in immediate reply.

A CT DLPDU may be sent on the link when the sending DLE holds a delegated token whose remaining allocated duration of token usage, C(RD), permits completion of the implied transaction — sending the CT DLPDU, which requires an immediate reply, and awaiting the TD reply DLPDU — prior to return of the token.

If the DLE holds a delegated token, and no additional use of that token after sending this DLPDU and awaiting its immediate reply is needed at that time, then the DLE may set the final-token-use subfield of the CT DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

After sending a CT DLPDU, the sending DLE shall monitor the local link for a reply as specified in 6.2.7.1. The permissible reply DLPDU is a TD DLPDU.

7.9.4 Receiving the CT DLPDU

A received CT DLPDU shall be treated as follows by the receiving DLE:

7.9.4.1 Actions required of all DLEs

none.

7.9.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.9.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.9.4.4 Additional actions required of the current LAS DLE

The DLE shall reply immediately with a TD DLPDU.

If the final-token-use subfield of the received CT DLPDU has the value FINAL, then the LAS DLE shall:

a) assume that the current use of the delegated token has terminated and that the scheduler token is again dominant on the local link;

b) treat that termination as if the token had been returned by an RT (see 7.17) DLPDU; and

c) resume active operation as the LAS.

7.10 Time Distribution (TD) DLPDU

A TIME DISTRIBUTION (TD) DLPDU is transmitted by the LAS DLE to enable the DLEs on the local link to coordinate and to synchronize the rates of advance of their senses of DL-time.

7.10.1 Structure of the TD DLPDU

Table 13 — Structure of TD DLPDUs

frame control	source NODE-address	parameters
0001 0F01	N	TD-p

7.10.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is URGENT;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU terminates use of a delegated token.

7.10.1.2 The address field shall consist of an explicit source NODE DL-address.

7.10.1.3 The parameter field shall be structured and encoded as specified in 8.6.

7.10.1.4 The user-data field shall be null.

7.10.2 Content of the TD DLPDU

The frame control field shall be encoded as specified in Table 13.

The address field shall specify the node designator of the LM DLE which is co-resident with the LAS DLE.

The TD-parameters field shall be encoded as specified in 8.6, and shall reflect the current values of the DLE's time-related variables and node-time counter at the moment of DLPDU formation, which shall be less than 1 s before DLPDU transmission.

The user-data field shall be null.

7.10.3 Sending the TD DLPDU

A TD DLPDU may be sent on the link when the sending DLE (the LAS)

a) has just received a reply token in a CT DLPDU, permitting a single transmission of a TD DLPDU; or

b) holds a scheduler token which is the dominant token on the local link, and is required to send a TD DLPDU as specified in 9.4.1.6, and the remaining allocated duration of token usage before the next scheduled activity permits completion of the DLPDU's transmission prior to expiration of the token.

If the TD DLPDU is sent as a reply to a received CT DLPDU, then the final-token-use subfield of the reply TD DLPDU shall have the same value as that in the requesting CT DLPDU.

If the time-synchronization class of the sending DLE is not NONE, then after transmitting the TD DLPDU, the LAS DLE shall schedule another transmission of a TD DLPDU as specified in 9.4.1.2.

Reception of a CT DLPDU by the LAS DLE shall cause the LAS DLE to reply immediately with a TD DLPDU, sent to all other DLEs on the local link.

7.10.4 Receiving the TD DLPDU

A received TD DLPDU shall be treated as follows by the receiving DLE:

7.10.4.1 Actions required of all DLEs

a) Each DLE that supports the variables, timers and counters defined in 5.7.1.19 through 5.7.1.26, other than the sending DLE, shall append to the received DLPDU the low-order 24 bits of the local node-time C(NT), known thereafter as N_R(NT), at which the TD DLPDU reception completed (that is, receipt of the END-OF-DATA or END-OF-DATA-AND-ACTIVITY PhIDU).

b) Each DLE other than the sending DLE shall process the received DLPDU as specified in 9.4.1.3.

7.10.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.10.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU. However, if the TD DLPDU was received at the port which is the bridge DLE's root port (that is, the port toward the root of the bridge spanning tree), then the DLE shall forward the DLPDU reception event to the bridge operation level (see 5.1.2).

7.10.4.4 Additional actions required of the current LAS DLE

If the DLE did not just send the DLPDU, then the DLE shall drop the scheduler token and deactivate its role as LAS; and shall inform local DL-management of the event.

If the final-token-use subfield of the received DLPDU has the value FINAL, then the LAS DLE shall

a) assume that the current use of the delegated token is terminated as if the token had been returned by an RT (see 7.17) DLPDU; and

b) assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.11 Round-Trip-Delay Query (RQ) DLPDU

A ROUND-TRIP-DELAY QUERY (RQ) DLPDU is sent from one DLE to another on the local link to initiate the measurement and computation of the round-trip delay intrinsic to their inter-communication. Its receipt results in the return of a complementary ROUND-TRIP-DELAY REPLY (RR) DLPDU completing the measurement.

7.11.1 Structure of the RQ DLPDU

Table 14 — Structure of RQ DLPDUs

frame control	destination address	source address	parameters
1100 0F00	N.0	N.0	RQ-p

7.11.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is NORMAL;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU, and either its expected immediate reply DLPDU or the appropriate link-idle timeout, terminates use of a delegated token.

7.11.1.2 The address field shall consist of two explicit SHORT DL-addresses, destination and source, in that order.

7.11.1.3 The parameter field shall be structured and encoded as specified in 8.7.

7.11.1.4 The user-data field shall be null.

7.11.2 Content of the RQ DLPDU

The frame control field shall be encoded as specified in Table 14.

The address field shall specify two NODE DL-addresses expressed in SHORT DL-address format, where the destination DL-address is

— the SHORT flat DL-address for the local LAS, 0400_{16} (see Annex A/A.3.2); or

— has the value formed by concatenating the intended receiver's node-id, V_R(TN), and an octet of zero, thus specifying that DLE's DL-support functions;

and the source DL-address has the value formed by concatenating the sender's node-id, V_S(TN), and an octet of zero, thus specifying the sending DLE's DL-support functions.

The RQ-parameters field shall be encoded as specified in 8.7, and shall reflect the value of the DLE's node-time counter, if maintained by the DLE, at a moment of DLPDU formation which is a constant time offset from the moment when the END-OF-DATA-AND-ACTIVITY PhIDU will be transmitted for the DLPDU. Otherwise the DLE shall use any value for this field.

The user-data field shall be null.

7.11.3 Sending the RQ DLPDU

An RQ DLPDU may be sent on the link when the sending DLE

a) holds a delegated token which is the dominant token on the local link, and when the remaining allocated duration of token usage, C(RD), permits completion of the implied transaction — sending the RQ DLPDU, which requires an immediate reply, and awaiting the RR reply DLPDU — prior to expiration of the token; or

b) holds a scheduler token which is the dominant token on the local link, and when both

   — the remaining allocated duration of token usage before the next scheduled activity; and

   — the remaining allocated duration of scheduler token usage for link maintenance, which is V(LTHT) octet-durations minus the amount of link capacity used for link maintenance during the current cycle of "circulating the token,"

   permit completion of the implied transaction — sending the RQ DLPDU, which requires an immediate reply, and awaiting the RR reply DLPDU — prior to expiration of the token.

The RQ DLPDU shall be sent at NORMAL priority.

If the DLE holds a delegated token, and no additional use of that token after sending this DLPDU and awaiting its immediate reply RR DLPDU is needed at that time, then the DLE may set the final-token-use subfield of the RQ DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

After sending an RQ DLPDU, the sending DLE shall monitor the local link for a reply as specified in 6.2.7.1. The permissible reply DLPDU is an RR DLPDU.

7.11.4 Receiving the RQ DLPDU

A received RQ DLPDU shall be treated as follows by the receiving DLE:

7.11.4.1 Actions required of all DLEs

If the destination NODE DL-address specified by the DLPDU designates the receiving DLE, then

a) when the receiving DLE's time-synchronization class (see 11.3(a)) is NONE and the DLE does not maintain even an estimated C(NT), then the DLE shall append to the received DLPDU the 24-bit value zero; or

b) else if (a) does not apply, then the DLE shall

1) append to the received DLPDU the low-order 24 bits of the local node-time, C(NT), at which the RQ DLPDU reception completed (that is, receipt of the END-OF-DATA or END-OF-DATA-AND-ACTIVITY PhIDU); and

2) adjust that 24-bit value to remove any systemic difference in the DLE-internal time delay between the DLE's transmit and receive paths, caused by known implementation considerations within the receiving real end system.

NOTES

1. This adjustment probably will vary inversely with the DLE's actual instantaneous rate of data transmission.

2. The objective of this adjustment is to ensure that V(MD), as computed in 9.4.1.5, is as close as possible to double the sum of the one-way transmission, propagation, and reception delays incurred in transmission of a TD DLPDU between the addressed source node and the local DLE.

In either case the receiving DLE shall initiate a reply within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of receipt of the RQ DLPDU. The reply DLPDU shall be an RR DLPDU with a destination DL-address equal to the source DL-address of the received RQ DLPDU, and with a source DL-address specifying the replying DLE's DL-support functions.

7.11.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.11.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.11.4.4 Additional actions required of the current LAS DLE

The LAS DLE shall act as specified in 6.2.7.2.

7.12 Round-Trip-Delay Reply (RR) DLPDU

A ROUND-TRIP-DELAY REPLY (RR) DLPDU is sent from one DLE to another on the local link to permit completion of the measurement and computation of the round-trip delay intrinsic to their inter-communication. It is only sent as an immediate reply to a received RQ DLPDU.

7.12.1 Structure of the RR DLPDU

Table 15 — Structure of RR DLPDUs

frame control	destination address	source address	parameters
1101 0F00	N.0	N.0	RR-p

7.12.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is NORMAL;

c) the length, number and type of DLPDU addresses; and

d) whether or not transmission of the current DLPDU terminates use of a delegated token.

7.12.1.2 The address field shall consist of two explicit SHORT DL-addresses, destination and source, in that order.

7.12.1.3 The parameter field shall be structured and encoded as specified in 8.8.

7.12.1.4 The user-data field shall be null.

7.12.2 Content of the RR DLPDU

The frame control field shall be encoded as specified in Table 15.

The address field shall specify two NODE DL-addresses expressed in SHORT DL-address format, where the destination DL-address is equal to the source DL-address received in the immediately prior RQ DLPDU, and the source DL-address has the value formed by concatenating the sender's node-id, V_S(TN), and an octet of zero, thus specifying the sending DLE's DL-support functions.

The RR-parameters field shall be encoded as specified in 8.8, and shall reflect the value of the DLE's node-time counter, if maintained by the DLE, at a moment of DLPDU formation which is a constant time offset from the moment when the END-OF-DATA-AND-ACTIVITY PhIDU will be transmitted for the DLPDU. Otherwise the DLE shall use the value zero for this field.

The user-data field shall be null.

7.12.3 Sending the RR DLPDU

An RR DLPDU may be sent on the link when the sending DLE has just received a reply token in an RQ DLPDU, permitting a single transmission of an RR DLPDU. The final-token-use subfield of the reply RR DLPDU shall have the same value as that in the requesting RQ DLPDU.

7.12.4 Receiving the RR DLPDU

A received RR DLPDU shall be treated as follows by the receiving DLE:

7.12.4.1 Actions required of all DLEs

If the destination NODE DL-address specified by the DLPDU designates the receiving DLE, and if the immediately prior DLPDU was an RQ DLPDU sent by the receiving DLE, then the DLE shall consider the prior transmission to have been error-free, and if the DLE maintains C(NT), then the DLE shall

a) append to the received DLPDU the low-order 24 bits of the local node-time, C(NT), at which the RR DLPDU reception completed (that is, receipt of the END-OF-DATA or END-OF-DATA-AND-ACTIVITY PhIDU);

b) adjust that 24-bit value to remove any systemic difference in the DLE-internal time delay between the DLE's transmit and receive paths, caused by known implementation considerations within the receiving real end system; and

NOTES

1. This adjustment probably will vary inversely with the DLE's actual instantaneous rate of data transmission.

2. The objective of this adjustment is to ensure that V(MD), as computed in 9.4.1.5, is as close as possible to double the sum of the one-way transmission, propagation, and reception delays incurred in transmission of a TD DLPDU between the addressed destination node and the local DLE.

c) DLE shall process the received DLPDU as specified in 9.4.1.5.

7.12.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.12.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.12.4.4 Additional actions required of the current LAS DLE

If the final-token-use subfield of the received DLPDU has the value FINAL, then the LAS DLE shall

a) assume that the current use of the delegated token is terminated as if the token had been returned by an RT **(see 7.17)** DLPDU; and

b) assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.13 Probe Node DL-address (PN) DLPDU

A PROBE NODE DL-ADDRESS (PN) DLPDU is used by the LAS DLE to probe for the existence of previously-unrecognized DLEs on the local link. A PN DLPDU creates and passes a reply token to any DLE(s) which may have their V(TN) (see 5.7.1.7) equal to the specified (configured but inactive) NODE DL-address. The only permissible reply is a PR DLPDU (see 7.14), which must be an immediate reply.

7.13.1 Structure of the PN DLPDU

Table 16 — Structure of PN DLPDUs

frame control	destination NODE-address	parameters
0010 0110	N	PN-p

7.13.1.1 The frame control field shall specify

a) the DLPDU's function; and

b) the length, number and type of DLPDU addresses.

7.13.1.2 The address field shall consist of an explicit destination NODE DL-address.

7.13.1.3 The parameter field shall be structured and encoded as specified in 8.9.

7.13.1.4 The user-data field shall be null.

7.13.2 Content of the PN DLPDU

The frame control field shall be encoded as specified in Table 16.

The address field shall specify the NODE DL-address which is being probed.

The Probe NODE DL-ADDRESS parameters (PN-parameters) field shall convey, as specified in 8.9, the then-current values for the local link of those DLE and PhE parameters necessary for a receiving DLE to configure itself and its associated PhE so that they can reply to this or a subsequent PN DLPDU.

The user-data fields shall be null.

7.13.3 Sending the PN DLPDU

The LAS on its own, at the beginning of each cycle of link maintenance-related transmissions, and possibly during what otherwise would be unoccupied periods of link capacity, probes configured but apparently-unused NODE DL-addresses by sending PN DLPDUs to those DL-addresses and awaiting a reply. Should such a reply occur, it is an indication that a new DLE has joined (or rejoined) the link, and the LAS then adds that DLE's NODE DL-address

— to its live list, V(LL), to be polled with PT DLPDUs; and

— to its token-circulation-list, V(TCL), if so requested in the responding PR DLPDU.

PN DLPDUs are not otherwise sent to NODE DL-addresses.

The LAS DLE shall send a PN DLPDU when probing NODE DL-addresses which are not associated with any known active DLEs on the local link. A PN DLPDU may be sent on the link by the LAS DLE when

a) the LAS DLE is using the link capacity allocated for link maintenance as specified in 10.2.1(b.1), 10.2.1(c.2); or

b) the scheduler token is the dominant token on the local link; and the remaining allocated duration of token usage before the next scheduled activity permits completion of a PN - PR transaction as specified in 10.2.1(b.3).

Before sending the PN DLPDU, the LAS DLE shall determine the next NODE DL-address to be probed by choosing, on alternate instances of sending the PN DLPDU, from the following two sets of NODE DL-addresses, as defined in Annex A/A.2.2, and cyclically in increasing numeric order within each set:

1) $\{ 10_{16} .. F7_{16} \} \; - \; V(LL) \; - \; \{ V(FUN) .. V(FUN) + V(NUN) - 1 \}$,

which is the set of configured bridge-class, link-master-class and basic-class DLE addresses, minus the set of active DLE addresses, and minus the set of excluded unused DLE addresses

2) $\{ F8_{16} .. FF_{16} \} \; - \; V(LL)$,

which is the set of visitor DLE addresses, plus the set of DLE addresses for non-visitor DLEs which do not know their proper DLE addresses, minus the set of active DLE addresses.

After sending a PN DLPDU, the LAS DLE shall monitor local link activity for a period of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations, waiting for a reply. After this interval, as soon as the link is inactive, the LAS DLE shall continue with its other activities. Even when a response has been received, the LAS shall wait until the end of the immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations, after the PN transmission, and then wait for the link to become inactive, before continuing with its next transaction, to provide for a potential second response from another DLE which has the same NODE DL-address.

Implementation NOTE — This requirement imposes timer considerations which differ subtly from all other uses of slot-time-based timers in the DL-protocol. For this timer, the timing action cannot stop when ACTIVITY or DATA is reported on the link, but rather must continue even after such reports.

If the LAS receives a DLPDU, which is not a PR DLPDU, during this monitoring interval, then the LAS DLE shall drop its scheduler token and inform local DL-management of the event.

7.13.3.1 Additional considerations

From the LAS DLE’s perspective, the maximum overhead link capacity required by any PN DLPDU and an upper bound on the maximum permitted responses may be pre-computed as

$$
\begin{aligned}
& 2 \times \text{framing-overhead} \\
+ \quad & 1 \times \text{PN-size} \\
+ \quad & 1 \times V(IRRD) \times V(ST) \\
+ \quad & 1 \times \text{maximum-PR-size} \\
+ \quad & 1 \times V(MID)
\end{aligned}
$$

7.13.4 Receiving the PN DLPDU

Receipt of a PN DLPDU by a newly-attached or newly-initialized DLE enables the DLE to learn those PhL and DLL configuration parameters which are essential for transmission on the local link.

Receipt of a PN DLPDU by the addressed DLE enables that DLE

— to indicate its presence and desired activity on the local link; and

— to begin the process of being included in the link’s other activities.

A received PN DLPDU shall be treated as follows by the receiving DLE:

7.13.4.1 Actions required of all DLEs

The DLE shall use the received PN DLPDU in the DLE-initialization procedures of 10.1.2.1 or 10.1.3 depending upon the DLE state. If those procedures require the receiving DLE to reply to the received PN DLPDU, then the DLE

a) shall assume a reply token;

b) shall initiate transmission of a PR DLPDU within a period of maximum-response-delay slot-times, $V(MRD) \times V(ST)$ octet-durations (as measured at the PN-receiving DLE); and

c) shall include a probe-response SPDU, as described in Annex B/B.3.2.1, in the user-data field of that PR DLPDU, and shall ensure that any randomly-chosen fields of that SPDU are chosen in a manner that is statistically independent of similar choices made by other DLEs.

7.13.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.13.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.13.4.4 Additional actions required of the current LAS DLE

If the DLE did not just send the DLPDU, then the DLE shall drop the scheduler token and deactivate its role as LAS; and shall inform local DL-management of the event.

7.14 Probe Response (PR) DLPDU

A PROBE RESPONSE (PR) DLPDU is sent to the LAS in response to the immediately-prior PROBE NODE DL-ADDRESS (PN) DLPDU, to convey probe-response information to the LAS.

7.14.1 Structure of the PR DLPDU

Table 17 — Structure of PR DLPDUs

frame control	**user data**
0010 0111	SPDU

7.14.1.1 The frame control field shall specify the DLPDU's function.

7.14.1.2 The address field shall be null.

7.14.1.3 The parameters field shall be null.

7.14.1.4 The user-data field shall consist of a higher-level probe-response SPDU.

NOTE — This small limit on the size of the probe-response SPDU facilitates probing of NODE DL-addresses during intervals within the link schedule which would otherwise be unusable.

7.14.2 Content of the PR DLPDU

The frame control field shall be encoded as specified in Table 17; the implicit value for the final-token-use subfield shall be FINAL.

The address field shall be null. The generic LAS NODE DL-address of 04 is implicitly addressed as the DLPDU's destination, and the destination address from the immediately prior PN DLPDU sent on the local link is implicitly the PR DLPDU's source address.

The parameters field shall be null.

The user data field shall convey a higher-level probe-response SPDU whose maximum size is limited as specified in 7.14.1.4.

7.14.3 Sending the PR DLPDU

A PR DLPDU shall result from receipt of a PN DLPDU. The PR DLPDU shall be sent as specified in 10 and 7.13.4.

The PR DLPDU shall not be forwarded by a bridge.

7.14.4 Receiving the PR DLPDU

A received PR DLPDU shall be treated as follows by the receiving DLE:

7.14.4.1 Actions required of all DLEs

none.

7.14.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.14.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.14.4.4 Additional actions required of the current LAS DLE

If the DLE had sent a PN DLPDU as the previous DLPDU on the link, then the DLE

a) shall report any higher-level SPDU conveyed by the received DLPDU, as if it had been received as DLS-user-data contained in a DT DLPDU addressed to the LAS's NODE DL-address and sent from the NODE DL-address which was contained in the last-sent PN DLPDU (see 7.13.3), and shall process the SPDU as specified in 10.2.3; and

b) shall assume that the scheduler token is again dominant on the local link, resume active operation as the LAS, and initiate transmission

1) within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of the end of reception of the PR DLPDU, as measured at the receiving DLE,

2) but not before the end of the observation interval following the PN DLPDU, as specified in 7.13.3.

If the DLE had sent a PN DLPDU as a previous DLPDU on the link, but another PR DLPDU was received after sending the PN DLPDU, and before reception of the current PR DLPDU, then the DLE shall ignore the current PR DLPDU and any contained SPDU.

If the DLE had not sent a PN DLPDU as the immediately prior non-PR DLPDU on the link, then the DLE shall drop the scheduler token and deactivate its role as LAS; and shall inform local DL-management of the event.

7.15 Pass Token (PT) DLPDU

A PASS TOKEN (PT) DLPDU is used to pass a delegated token from the DLE functioning as an LAS to a DLE on the local link. By doing this repeatedly, the LAS DLE provides a delegated token which "circulates" successively, usually in NODE DL-address order, to all active DLEs on the local link which are included in the link's token-circulation list, V(TCL) (see 5.7.5.3).

NOTE — This usage also supports migration from previously-existing national standards.

This DLPDU provides the receiving DLE with the right to initiate DL-transactions for a period of time specified in the delegating DLPDU.

7.15.1 Structure of the PT DLPDU

Table 18 — Structure of PT DLPDUs

frame control	destination NODE-address	parameters
0011 0FPP	N	DD-p

7.15.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the minimum required priority of token use;

c) the phase (initial or continuation) of the token delegation; and

d) the length, number and type of DLPDU addresses.

7.15.1.2 The address field shall consist of an explicit destination NODE DL-address.

7.15.1.3 The parameter field shall be structured and encoded as specified in 8.10.

7.15.1.4 The user-data field shall be null.

7.15.2 Content of the PT DLPDU

The frame control field shall be encoded as specified in Table 18. The value of the priority subfield determines the minimum permitted priority of DLPDU which may be transmitted by the receiving DLE. The value of the final-use subfield shall be interpreted generally as specified in 6.2.1.2(c), and specifically as:

RESTART —this is the initial token delegation within the current cycle of "circulating the token," and so any repetitively-scheduled transactions should be restarted;

CONTINUE —this is a subsequent (that is, secondary) token delegation within the current cycle of "circulating the token," and so the previously-initiated sequence of queued transactions should be continued.

The address field shall specify the NODE DL-address to which the token is being delegated.

The Delegation-Duration-parameter (DD-parameter) field shall convey, as specified in 8.10, the duration for which the token is being delegated to service Q(US), measured in octet-durations. Its permitted values are 0 to 65 000, measured in units of the transmission-duration of one octet (that is, in octet-durations).

NOTE — The lower bound of zero is required to permit probing of node DL-addresses for DLEs which are included in the local link's live-list, V(LL), but not in the local link's token circulation list, V(TCL). The minimum practical value for use of this duration is about 16.

A duration of zero shall be used only when the addressed DLE's NODE DL-address is in the live list, V(LL), but not in the list of DLEs, V(TCL), to be polled with PT DLPDUs. In this case the addressed DLE indicates its continued presence by responding with an RT or RI DLPDU.

The user-data field shall be null.

7.15.3 Sending the PT DLPDU

The LAS DLE sends a PT DLPDU to "circulate the token," usually in NODE DL-address order, through all of the active DLEs on the local link.

7.15.3.1 Determination of the PT DLPDU fields and related "token-rotation" parameters

a) A PT DLPDU may be sent on the link

— when the sending DLE (the LAS) holds a scheduler token which is the dominant token on the local link;

— when the number of octet-durations of link capacity remaining until the next scheduled activity permits sending the PT DLPDU, plus the minimal required use by the receiving DLE, plus return of the token in an RI DLPDU or recovery of the token if required; and

— when the sending DLE does not need to send some other DLPDU.

b) The overhead in delegating the token, which is not included in the delegated link capacity specified in the DD-parameters of the PT DLPDU, but which is factored into the determination of whether to send the PT DLPDU, is computed as the sum of

1) the link capacity required to send the PT DLPDU;

2) V(IRRD) × V(ST) octet-durations for token acceptance and commencement of initial use;

3) the link capacity required for the delegated-token-holder to send an RI DLPDU terminating token usage; and

4) the larger of

i) the minimum inter-DLPDU delay, V(MID) octet-durations (see 5.7.1.12), which follows the RI DLPDU; or

ii) any implementation-dependent period, measured in units of link capacity, required to recover the token upon expiration of the link capacity actually delegated.

c) The maximum number of octet-durations of link capacity which can be delegated is computed as the smaller of

1) 65 000 octet-durations; or

2) the number of octet-durations of link capacity remaining until the next scheduled activity minus the overhead in delegating the token specified in (b), with the result reduced to account for possible differences in the two DLEs' data rates (that is, multiplied by the minimum possible ratio between the LAS DLE's and receiving DLE's Ph-data rates, as reported in 5.4.1.1.

d) The minimum number of octet-durations of link capacity which a DLE requires shall be computed as follows:

1) zero, when the DLE's NODE DL-address is not a member of the set V(TCL), indicating that the DLE does not require use of the token;

2) V(DMDT) (see 5.7.5.7), when (1) does not apply, and when

— the PT DLPDU's final-use subfield specifies RESTART; or

— the last token sent by a PT DLPDU to the same DL-address was not returned by an RI DLPDU; or

3) the value from the DD-parameters of the RI DLPDU which returned the last token sent by a PT DLPDU to the same DL-address, when (1) and (2) do not apply.

e) The maximum number of octet-durations of link capacity which may be delegated to a DLE shall be computed as follows:

1) zero, when the DLE's NODE DL-address is not a member of the set V(TCL), indicating that the DLE does not require use of the token;

2) the member of the array V(MTHA) corresponding to the DLE's NODE DL-address, when (1) does not apply, and when the PT DLPDU's final-use subfield specifies RESTART; and

3) the member of the array V(RTHA) corresponding to the DLE's NODE DL-address, when (1) and (2) do not apply.

NOTE — The link capacity actually used by token-returning RI and RT DLPDUs is not included in this computation, since that link capacity is allocated to the overhead of the token-passing process.

f) The token shall be delegated in ascending order of NODE DL-addresses to those DLE NODE DL-addresses represented in V(LL), beginning with the lowest-numbered such NODE DL-address. When a DLE has just assumed the scheduler token (by sending a CL DLPDU or receiving a TL DLPDU) and becomes the LAS, it shall commence token-passing with this lowest-numbered NODE DL-address, and shall restart token-passing with the lowest-numbered DLE after all other DLE's represented in the set V(LL) have terminated their token usage

— by specifying a final-use of FINAL; or

— by being unable to use the token without exceeding the maximum number of octet-durations of link capacity determined in (e).

A DLE can, but need not, be temporarily bypassed in this ascending order while its requirements, as computed in (d), exceed the link capacity momentarily available for token delegation, as computed in (c). Any such bypassed DLE shall be delegated its remaining allocated link capacity before a PT DLPDU whose final-use field specifies RESTART is again delegated to that DLE.

It is a protocol error to specify any DL-address other than a valid NODE DL-address, or to explicitly address a DLE which is known to be not active on the local link.

g) The actual period of token delegation shall be computed to be the lesser of the values computed in (c) and (e), and the delegation shall occur only if that period is at least as great as the value computed in (d).

h) The initial PT DLPDU sent to each DLE on the link during one cycle of "circulating the token" shall specify RESTART in its final-use field; all subsequent PT DLPDUs sent to the same DLE during the same cycle, if any, shall specify CONTINUE in that final-use field. When RESTART is specified, the DLE also shall initialize the relevant member of V(RTHA) to the value of the corresponding member of V(MTHA).

j) The actual period of token rotation, V(ATRT), shall be measured by the LAS DLE as the interval between successive occurrences of the LAS DLE's sending the PT DLPDU, with a token-use subfield specifying RESTART (see 7.15.2), to the lowest-numbered NODE DL-address represented in V(LL).

k) The DLL priority specified in the PT DLPDU shall be determined before each cycle of "circulating the token" as follows:

1) After the LAS DLE has just assumed the scheduler token (by sending a CL DLPDU or receiving a TL DLPDU), the priority shall be NORMAL.

2) If (1) does not apply, and the just-computed V(ATRT) was greater than V(TTRT), then

— DL-management shall be notified of the event; and

— the token priority shall be increased to the next higher level of urgency, if possible.

3) If (1) and (2) do not apply, so that the just-computed V(ATRT) was less than or equal to V(TTRT), then the token priority shall be decreased to the next lower level of urgency, if possible.

NOTE — The inclusion of hysteresis or "learning" in this priority adjustment is an area of future study.

Once determined, the same DLL priority shall be included in all PT DLPDUs transmitted during that cycle of "circulating the token."

7.15.3.2 Sending the PT DLPDU and monitoring the DLE to which the token is delegated

When sending the PT DLPDU, the LAS DLE shall record the destination address from the DLPDU as the delegation-address, V(DTA) (see 5.7.5.1), for potential DL-management use in case the token is lost and not returned, and for potential association with any received RI or RT DLPDU.

After sending a PT DLPDU, the LAS DLE shall monitor the local link for a period of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations, waiting for a response:

a) If a PhL-indication (see 5.4.3) reporting DATA is received, then the DLE shall proceed as in (c).

b) If (a) does not apply, and the monitoring period of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations expires, then

1) If the bus is not active at that moment (that is, the last-received PhL-indication reported END-OF-ACTIVITY), then the DLE shall proceed as in (d).

2) If (1) does not apply, implying that the link is still active at that moment (that is, the last-received PhL-indication reported START-OF-ACTIVITY), then

A) Unless (b.2.B) applies, the DLE shall monitor the local link for a period of one additional slot-time, V(ST) octet-durations, waiting for a PhL-indication:

i) If a PhL-indication reporting DATA is received, then the DLE shall proceed as in (c).

ii) If a PhL-indication reporting END-OF-ACTIVITY is received, and (i) does not apply, then the DLE shall stop further monitoring and proceed as in (d).

NOTE — Without provision (ii), ill-timed low-level noise can cause the LAS DLE to not retry the token pass, resulting in omitting a DLE from the current cycle of circulating the token. Provision (ii) enables the LAS DLE to distinguish probable noise events from actual transmissions.

iii) If neither (i) nor (ii) applies, and the monitoring period expires before a PhL-indication is received, then the DLE shall proceed as in (c).

B) Implementations based on hardware that was designed specifically to implement this International Standard, where such hardware was demonstrable on or before 31 December 1995 alternatively may just proceed as in (c).

NOTE — After this document achieved initial ACDV status, implementors were encouraged to develop chips to assist in evaluating this complex protocol. As a consequence, it was found desirable to improve the noise rejection characteristics of the token passing process, and the text (b.2.A) and (c.2), which would require changes in those existing chips, was the result. (B) grandfathers the noise rejection approach of the first ACDV for those early implementations, and only for those early implementations.

c) Until the delegated token is returned, the LAS DLE shall continually monitor local link activity, and if it observes

1) a period of token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, of continuous non-activity; or

2) unless (b.2.B) applies, a period of token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, during which any continuous period of activity (that is, between a PhL-indication reporting START-OF-ACTIVITY and the next following PhL-indication reporting END-OF-ACTIVITY or END-OF-DATA-AND-ACTIVITY) both

i) did not result in a PhL-indication (see 5.4.3) reporting DATA, and

ii) had a duration of less than one to two V(ST) octet-durations

NOTE — The vagueness in the extent of permissible activity duration permits some economy of implementation of the measurement.

then it shall

— assume that the scheduler token is dominant on the local link;

— inform local DL-management of the event; and

— commence transmission on the link.

d) If the monitoring period of immediate-response-recovery-delay slot-times,

V(IRRD) × V(ST) octet-durations expires and (a) does not apply; or, where (b.2.B) does not apply, the additional monitoring period of one slot-time, V(ST) octet-durations, expires and (c) does not apply; then the LAS DLE shall retry sending the PT DLPDU and monitoring the local link, up to the number of times specified as the maximum-retry-count, V(MRC) (see 5.7.1.5), of the local link, each time using as the value for its duration parameter the smaller of

1) the value originally-sent; or

2) the number of octet-durations of link capacity remaining until the next scheduled activity minus the overhead in delegating the token specified in 7.15.3.1(b), with the result reduced to account for possible differences in the two DLEs' data rates as specified in 7.15.3.1(c.2),

provided that the retry and attempted re-delegation shall occur only if that period is at least as great as the value computed in 7.15.3.1(d).

This retry shall commence within token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, of the beginning of the current period of link non-activity.

If all attempted retries are unsuccessful, then the LAS DLE shall

— inform local DL-management of the event; and

NOTE — DLEs with node DL-addresses in the set { $F8_{16}$ .. FF_{16} } are expected to terminate operation by dropping out of the token circulation process. Thus, DL-management should not treat such occurrences as evidence of DLE or local-link malfunction.

— start the next transmission within token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, of the beginning of the current period of link non-activity.

After three such event reports for any specific NODE DL-address within three cycles of "circulating the token," DL-management shall remove that NODE DL-address from the LAS's local-link token-circulation-list, V(TCL), and local-link live-list, V(LL).

7.15.4 Receiving the PT DLPDU

Receipt of a PT DLPDU by the addressed DLE enables the DLE to emulate a "circulated token" as found in previous token-passing bus standards, such as ISO/IEC 8802-4 and IEC 955, by enabling transmission of DLPDUs at the specified and higher priority either

— until all pending transmissions have occurred; or

— until the portion of the local-link's transmission capacity which was allocated by the DD-parameters of the received PT DLPDU requires return of the delegated token.

A received PT DLPDU shall be treated by the receiving DLE as follows:

7.15.4.1 Actions required of all DLEs

If the destination DL-address specified by the DLPDU designates the NODE DL-address of the receiving DLE, then if the receiving DLE, due to its construction, does not use the delegated token, then the receiving DLE shall initiate transmission of an RT DLPDU within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of the end of reception of the PT DLPDU, as measured at the receiving DLE; else the receiving DLE shall

a) copy the value of the duration parameter from the DLPDU into its local remaining-duration down-counter, C(RD), if implemented;

b) assume the delegated token;

c) initiate transmission within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of the end of reception of the PT DLPDU, as measured at the receiving DLE; and

d) repeatedly employ the following selection criteria until the delegated token is returned to the LAS, each time initiating transmission within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of the end of the prior transmission (unless that was a CA, CD, CT, ED or RQ transmission), as measured at the PT-receiving DLE.

The counter C(RD), if implemented, shall decrement and be managed as specified in 5.7.1.11.

Transactions of the priority specified in the delegating PT DLPDU, or of higher priority

1) shall be initiated successively;

2) beginning as follows:

— starting from the first transaction and including all REPETITIVE sequences if the token-use subfield of the received PT DLPDU specified RESTART; or

— continuing from the previously-initiated transaction and thus excluding all REPETITIVE sequences which have been executed since the last PT DLPDU specifying RESTART was received if the token-use subfield of the received PT DLPDU specified CONTINUE;

3) subject to the constraint that the DLE shall limit its token use to a sequence of transactions which is guaranteed to complete within the duration specified in the DD-parameter of the received PT DLPDU.

7.15.4.1.1 Selection of the next transaction to be executed

If the DLE has a need to send a CT DLPDU, the DLE shall transmit a CT DLPDU and await its immediate reply. Otherwise, the receiving DLE repeatedly shall select the next member, if any, of the DLE's unscheduled-service queue, Q(US), whose priority is not less than the priority of the received PT, and shall apply all of the remainder of this sub-clause, as appropriate:

a) If that next member is a reference to a DL-address user-request queue, Q_A(UR), then

1) If the member is a reference which signals the need to compel a DLSDU transmission from the remote correspondent of a local peer or subscriber DLCEP (see 5.7.1.17(a.1)), then the DLE shall

i) transmit a CD DLPDU to the remote DLC endpoint, with SD-parameters conveying the state of the local DLCEP to the remote DLCEP, as specified in 9.2.2.1 and 9.2.2.4;

ii) await an immediate reply; and

NOTE — The maximum DLPDU size of that immediate reply can be determined solely on the basis of the negotiated DLCEP parameters.

iii) delete the referencing member from the DLE's unscheduled-service queue, Q(US).

2) If the member is a reference which signals the need to compel an instance of unitdata exchange with a specified DLSAP-address at a specified service priority (see 5.7.1.17(a.2)), then the DLE shall

i) transmit an ED or CD DLPDU to that remote DLSAP-address, at the specified service priority, as appropriate, as specified in 9.2.2.1 and 9.2.2.4;

ii) await an immediate reply; and

NOTE — The maximum DLPDU size of that immediate reply can be determined solely on the basis of the negotiated DLCEP parameters.

iii) delete the referencing member from the DLE's unscheduled-service queue, Q(US).

3) If the member is a reference which signals the need to transmit the contents of a sending buffer which is bound to a local DLCEP (see 5.7.1.17(a.3)), then the DLE shall

i) transmit the DLPDU appropriate to the state of the DLCEP; and

ii) if the resultant transmitted DLPDU completes the transmission of a DLSDU (so that no segments of that DLSDU remain to be transmitted), then the DLE shall

— notify the DLS-user of the request's completion, if appropriate; and

— delete the referencing member from the DLE's unscheduled-service queue, Q(US).

4) If the member is a reference which signals the need to send a DLSDU from the sending queue which is bound to a local DLCEP or "DLSAP-address" (see 5.7.1.17(a.4)), or to send a DLPDU from the DLE's NODE DL-address, then the DLE shall select a member of the referenced Q_A(UR), of the specified or higher priority, which has been released for transmission and which requires full or partial transmission or retransmission, as follows:

— either the first member of the second partition of Q_A(UR); or

— the first member of the first partition of Q_A(UR) which has one or more segments marked for retransmission.

NOTE — Due to potential concurrency in real implementations, the set of segments so marked may not reflect status information from DLPDUs recently received at the DLCEP.

If such a member exists, then

i) the DLE shall transmit the DLPDU with a segment of the DLSDU appropriate to the transmission segment status, $V_{C,K}$ (SS) (see 5.7.4.8), if applicable, of that member of that Q_A(UR); and

ii) if the resultant transmitted DLPDU completes the transmission of a DLSDU (so that no segments of that DLSDU remain to be transmitted), then the DLE shall

— advance that member of Q_A(UR) to the already-sent partition, or remove it, as appropriate to the type of member;

— notify the DLS-user of the request's completion, if appropriate; and

— delete the referencing member from the DLE's unscheduled-service queue, Q(US).

If no such member of the referenced DL-address request queue, Q_A(UR), exists, then

iii) if the DL-address is a DLCEP-address of a local peer or publisher DLCEP, then the DLE shall

A) transmit a DT or RC DLPDU, as appropriate for the state of the DLCEP, with a null user-data field, to convey the state of the DLCEP to the correspondent peer or subscriber DLCEP(s); and

B) delete the referencing member from the DLE's unscheduled-service queue, Q(US).

iv) otherwise, when (iii) does not apply, then the DL-address is not a DLCEP-address and the DLE shall repeat the selection process.

b) If that next member is a reference to a REPETITIVE sequence (see 9.4.3.1(d)), then

1) if there are one or more active members of the sequence (that is, not excluded by the most recent DL-SUBSET-SEQUENCE request, if any, specifying the sequence); and

2) if one or more of those active members has not been executed since the DLE last received a PT DLPDU specifying RESTART,

then the DLE shall

i) insert a reference to the next active member of that sequence on Q(US) immediately before the reference to the sequence (with the result that the newly-inserted reference will be deleted at completion of the just-scheduled action); and

ii) continue as in the preceding case (a), as determined by the type of the just-inserted member of the sequence.

NOTE — At completion of the just-scheduled action, the next member of Q(US) will be the reference to the REPETITIVE sequence, and the above procedure will continue until all active members of the REPETITIVE sequence have been processed.

3) otherwise, when (1) does not apply or (2) does not apply, then the DLE shall repeat the selection process.

c) If that next member is a reference to a sequence (see 9.4.3.1(a)) which is not a REPETITIVE sequence, and if the sequence consists of a single element, then the DLE shall

1) replace the reference to the sequence on Q(US) with the reference to the single element, so that the same member of Q(US) now is the element reference;

2) issue a DL-SCHEDULE-SEQUENCE confirm primitive if it has not already been confirmed;

3) delete the sequence; and

4) continue as in the preceding case (a), as determined by the type of the sole element of the selected and just-deleted sequence.

All other cases are erroneous and are not permitted.

7.15.4.1.2 Additional considerations

The following considerations also apply:

i) If there are no additional members, of the priority specified in the received PT DLPDU, or of any higher priority, in the DLE’s unscheduled-service queue, Q(US), and if immediate retry of the currently-selected transaction is not possible, then the final-token-use subfield in that non-RT DLPDU can be set to the value FINAL before transmission, thereby returning the delegated token to the LAS DLE at the end of the current transaction.

ii) If there are no members, of the priority specified in the received PT DLPDU, or of any higher priority, in the DLE’s unscheduled-service queue, Q(US), and the delegated token was not returned by setting a final-token-use subfield to the value FINAL in the last DLPDU transmitted by the DLE, then the DLE shall return the delegated token by transmitting an RT DLPDU.

iii) If at any time

A) the remaining duration of token usage — as indicated by the remaining-duration down-counter, C(RD), or its functional replacement (see 5.7.1.1) — is inadequate to permit any further use by the receiving DLE; and

B) the DLE's unscheduled-service queue, Q(US), contains additional members whose priority is equal to or higher than the priority specified in the last-received PT DLPDU,

then the receiving DLE shall return the delegated token to the LAS by sending an RI DLPDU specifying the minimum delegation interval for token use required when the token is next returned.

NOTE — An RI DLPDU, which is sent to terminate use of a token delegated by a PT DLPDU, indicates to the receiving LAS that additional service is needed and that another PT DLPDU, with a final-token-use subfield specifying CONTINUE, should be sent to the DLE during the current cycle of "circulating the token" if the total link capacity allocated to the DLE by the DLE'th member of V(MTHA) (see 5.7.5.10) so permits.

iv) If none of (i) through (iii) applies, then the DLE shall transmit the non-RT non-RI DLPDU with a final-token-use subfield value of NOT-FINAL.

7.15.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.15.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.15.4.4 Additional actions required of the current LAS DLE

If the DLE did not just send the DLPDU, then the DLE shall drop the scheduler token and deactivate its role as LAS; and shall inform local DL-management of the event.

7.16 Execute Sequence (ES) DLPDU

An EXECUTE SEQUENCE (ES) DLPDU is used to pass a delegated token from the DLE functioning as an LAS to a DLE on the local link which has previously requested or scheduled local link capacity, or for which such a schedule has been configured by DL-management.

NOTE — This usage supports migration from previously-existing national standards.

This DLPDU provides the receiving DLE with the right to initiate DL-transactions for a period of time, specified in the delegating DLPDU; the destination DL-address of that DLPDU is used to determine which set of transactions to initiate.

7.16.1 Structure of the ES DLPDU

Table 19 — Structure of ES DLPDUs

format	frame control	destination DLSEP-address	parameters
1L	1000 1F00	HL.N.S	DD-p
1S	1000 0F00	N.S	DD-p

7.16.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the phase (initial or continuation) of the sequence execution; and

c) the length, number and type of DLPDU addresses.

7.16.1.2 The address field shall consist of an explicit destination DL-address.

7.16.1.3 The parameter field shall be structured and encoded as specified in 8.10.

7.16.1.4 The user-data field shall be null.

7.16.2 Content of the ES DLPDU

The frame control field shall be encoded as specified in Table 19. The value of the final-use subfield shall be interpreted generally as specified in 6.2.1.2(c), and specifically as:

RESTART — this is the initial token delegation within the current cycle of scheduled sequence execution, and so the indicated sequence and any repetitively-scheduled transactions should be restarted;

CONTINUE — this is a subsequent (that is, secondary) token delegation within the current cycle of scheduled sequence execution, and so the previously-initiated sequence of scheduled transactions should be continued at the point where execution was last suspended and the ES DLPDU was re-requested (see 7.18.3).

The address field shall specify the DLSEP-address to which the token is being delegated. For format 1L, the address shall be LONG; for format 1S it shall be SHORT.

The Delegation-Duration-parameter (DD-parameter) field shall convey, as specified in 8.10 the duration for which the token is being delegated to execute the addressed scheduled sequence, measured in units of the transmission-period of one octet on the local link (that is, eight times the local link's nominal bit period). It has the range 0 to 65 000 octet-durations.

The user-data field shall be null.

7.16.3 Sending the ES DLPDU

The LAS DLE sends an ES DLPDU when so instructed by the local link's current schedule, or in response to requests to extend the time allocated for an earlier scheduled (and executed) token delegation.

7.16.3.1 Sending the ES DLPDU as instructed by the local link's current schedule

An ES DLPDU may be sent on the link when the sending DLE (the LAS) holds a scheduler token which is the dominant token on the local link, and the link's schedule requires transmission of the ES DLPDU. The final-use subfield of such an ES DLPDU shall be dictated by the schedule.

7.16.3.2 Sending the ES DLPDU in response to earlier requests for extensions of scheduled token delegation

An ES DLPDU shall only be sent as a response to an RI DLPDU as specified in 10.2.2. If the LAS DLE sends an ES DLPDU which is not dictated by the link's current schedule, then the following procedures shall apply:

a) An ES DLPDU may be sent on the link when the sending DLE (the LAS) holds a scheduler token which is the dominant token on the local link, and when the number of octet-durations of link capacity remaining until the next scheduled activity permits sending the ES DLPDU, plus the minimal required use by the receiving DLE, plus return of the token in an RI DLPDU or recovery of the token if required. The final-use subfield of such an ES DLPDU shall specify CONTINUE.

b) The overhead in delegating the token, which is not included in the delegated link capacity specified in the DD-parameters of the ES DLPDU, but which is factored into the determination of whether to send the ES DLPDU, is computed as the sum of

1) the link capacity required to send the ES DLPDU;

2) V(IRRD) × V(ST) octet-durations for token acceptance and commencement of initial use;

3) the link capacity required for the delegated-token-holder to send an RI DLPDU terminating token usage; and

4) the larger of

i) the minimum inter-DLPDU delay, V(MID) octet-durations (see 5.7.1.12), which follows the RI DLPDU; or

ii) any implementation-dependent period, measured in units of link capacity, required to recover the token upon expiration of the link capacity actually delegated.

c) The maximum number of octet-durations of link capacity which can be delegated is computed as the smaller of

1) 65 000 octet-durations; or

2) the number of octet-durations of link capacity remaining until the next scheduled activity minus the overhead in delegating the token specified in (b), with the result reduced to account for possible differences in the two DLEs' data rates (that is, multiplied by the minimum possible ratio between the LAS DLE's and receiving DLE's Ph-data rates, as reported in 5.4.1.1).

d) The actual period of token delegation shall be the value returned in the DD-parameters of the RI DLPDU which terminated the prior interval of delegated token usage at the same DL-address.

7.16.3.3 Monitoring the DLE to which the token is delegated and additional considerations

It is a protocol error to specify any DL-address other than a valid DLSEP-address, or to explicitly or implicitly address a DLE not active on the local link. However, an LAS which receives a DLSEP-address from another DLE in a scheduling-related SPDU may presume that such an address is valid.

When sending the ES DLPDU, the LAS DLE shall record the destination address from the DLPDU as the delegation-address, V(DTA) (see 5.7.5.1), for potential DL-management use in case the token is lost and not returned, and for potential association with any received RI or RT DLPDU.

After sending an ES DLPDU, the LAS DLE shall monitor the local link for a period of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations, waiting for a response:

a) If a PhL-indication (see 5.4.3) reporting DATA is received, then the DLE shall proceed as in (c).

b) If (a) does not apply, and the monitoring period of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations expires, then

1) If the bus is not active at that moment (that is, the last-received PhL-indication reported END-OF-ACTIVITY), then the DLE shall proceed as in (d).

2) If (1) does not apply, implying that the link is still active at that moment (that is, the last-received PhL-indication reported START-OF-ACTIVITY), then

A) Unless (b.2.B) applies, the DLE shall monitor the local link for a period of one additional slot-time, V(ST) octet-durations, waiting for a PhL-indication:

i) If a PhL-indication reporting DATA is received, then the DLE shall proceed as in (c).

ii) If a PhL-indication reporting END-OF-ACTIVITY is received, and (i) does not apply, then the DLE shall stop further monitoring and proceed as in (d).

NOTE — Without provision (ii), ill-timed low-level noise can cause the LAS DLE to not retry the token pass, resulting in omitting a DLE from the current cycle of schedule execution. Provision (ii) enables the LAS DLE to distinguish probable noise events from actual transmissions.

iii) If neither (i) nor (ii) applies, and the monitoring period expires before a PhL-indication is received, then the DLE shall proceed as in (c).

B) Implementations based on hardware that was designed specifically to implement this International Standard, where such hardware was demonstrable on or before 31 December 1995 alternatively may just proceed as in (c).

NOTE — After this document achieved initial ACDV status, implementors were encouraged to develop chips to assist in evaluating this complex protocol. As a consequence, it was found desirable to improve the noise rejection characteristics of the token passing process, and the text (b.2.A) and (c.2), which would require changes in those existing chips, was the result. (B) grandfathers the noise rejection approach of the first ACDV for those early implementations, and only for those early implementations.

c) Until the delegated token is returned, the LAS DLE shall continually monitor local link activity, and if it observes

1) a period of token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, of continuous non-activity; or

2) unless (b.2.B) applies, a period of token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, during which any continuous period of activity (that is, between a PhL-indication reporting START-OF-ACTIVITY and the next following PhL-indication reporting END-OF-ACTIVITY or END-OF-DATA-AND-ACTIVITY) both

i) did not result in a PhL-indication (see 5.4.3) reporting DATA, and

ii) had a duration of less than one to two V(ST) octet-durations

NOTE — The vagueness in the extent of permissible activity duration permits some economy of implementation of the measurement.

then it shall

— assume that the scheduler token is dominant on the local link; and

— inform local DL-management of the event; and

— commence transmission on the link.

d) If the monitoring period of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations expires and (a) does not apply; or, where (b.2.B) does not apply, the additional monitoring period of one slot-time, V(ST) octet-durations, expires and (c) does not apply; then the LAS DLE shall retry sending the ES DLPDU and monitoring the local link, up to the number of times specified as the maximum-retry-count, V(MRC) (see 5.7.1.5), of the local link, each time using as the value for its duration parameter the value originally-sent, and the re-delegation shall occur only if that period is at least as great as the value specified in 7.16.3.1 or 7.16.3.2(d).

This retry shall commence within token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, of the beginning of the current period of link non-activity.

If all attempted retries are unsuccessful, then the LAS DLE shall

— inform local DL-management of the event; and

— start the next transmission within token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, of the beginning of the current period of link non-activity.

7.16.4 Receiving the ES DLPDU

Receipt of an ES DLPDU by the addressed DLE enables the DLE to execute all or part of a previously-scheduled sequence whose initiation is synchronized by the LAS, either

— until the sequence has been completed; or

— until the portion of the local-link's transmission capacity which was allocated by the DD-parameters of the received ES DLPDU requires return of the delegated token.

NOTE — This latter situation can occur

A) on sequences which permit interruption;

B) on sequences which permit dynamic appending of DL-COMPEL-SERVICE requests; and

C) when retransmission of the initiating ES DLPDU, or possibly of another DLPDU within the defined sequence, caused link capacity beyond that allocated to be needed.

A received ES DLPDU shall be treated by the receiving DLE as follows:

7.16.4.1 Actions required of all DLEs

7.16.4.1.1 When addressed to a DL-address of the DLE other than a DLSEP-address

If the destination DL-address specified by the DLPDU designates a DL-address of the receiving DLE other than a DLSEP-address, then the DLE shall:

a) inform local DL-management of the event; and

b) initiate transmission within a period of maximum-response-delay slot-times,

V(MRD) × V(ST) octet-durations, as measured at the receiving DLE, of an RT DLPDU.

NOTE — The receiving DLE is rejecting the received ES DLPDU.

7.16.4.1.2 When addressed to a DLSEP-address of the DLE

If the destination DL-address specified by the DLPDU designates a DLSEP-address of the receiving DLE, then the receiving DLE shall

a) copy the value of the duration parameter from the DLPDU into its local remaining-duration down-counter, C(RD);

b) assume the delegated token;

c) initiate transmission within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of the end of reception of the ES DLPDU, as measured at the receiving DLE; and

d) repeatedly employ the following selection criteria until the delegated token is returned to the LAS, each time initiating transmission within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, of the end of the prior transmission (unless that was a CA, CD, CT or ED transmission), as measured at the ES-receiving DLE.

The counter C(RD) shall decrement and be managed as specified in 5.7.1.11.

Transactions in the sequence associated with the designated DLSEP-address

1) shall be initiated successively;

2) beginning as follows:

— starting with the initial transaction of the sequence if the token-use subfield of the received ES DLPDU specified RESTART; or

— continuing from the point at which the sequence was previously suspended if the token-use subfield of the received ES DLPDU specified CONTINUE; and

3) subject to the constraint that the worst-case link capacity required to complete the transaction be less than the then-current value of the remaining-duration down-counter, C(RD).

7.16.4.1.2.1 Selection of the next transaction to be executed

The receiving DLE shall execute the transactions in the sequence as follows:

If that next member of the sequence is a reference to a DL-address user-request queue, Q_A(UR), then

a) If the member is a reference to the remote correspondent of a local peer or subscriber DLCEP (see 9.4.2.1(b.1.i)), then the DLE shall:

1) transmit a CD DLPDU to the remote DLC endpoint, with SD-parameters conveying the state of the local DLCEP to the remote DLCEP, as specified in 9.2.2.1 and 9.2.2.4;

2) await an immediate reply; and

NOTE — The maximum DLPDU size of that immediate reply can be determined solely on the basis of the negotiated DLCEP parameters.

3) advance to the next element of the sequence which was addressed by the received ES DLPDU.

b) If the member is a reference which indicates the need to perform a unitdata exchange with a specified DLSAP-address at a specified service priority (see 9.4.2.1(c.1.i)), then the DLE shall

1) transmit an ED or CD DLPDU to that remote DLSAP-address, at the specified service priority, as appropriate, as specified in 9.2.2.1 and 9.2.2.4;

2) await an immediate reply; and

NOTE — The maximum DLPDU size of that immediate reply can be determined solely on the basis of the negotiated DLCEP parameters.

3) advance to the next element of the sequence which was addressed by the received ES DLPDU.

Otherwise, the DLE shall select the first member of the referenced Q_A(UR) which has been released for transmission and which requires full or partial transmission or retransmission.

c) If such a member exists, then

1) the DLE shall transmit the DLPDU appropriate to the state of that member of that Q_A(UR); and

2) if the resultant transmitted DLPDU completes the transmission of a DLSDU (so that no segments of that DLSDU remain to be transmitted), then the DLE shall

— advance that member of Q_A(UR) to the already-sent partition, or remove it, as appropriate to the type of member;

— notify the DLS-user of the request’s completion, if appropriate; and

— advance to the next element of the sequence which was addressed by the received ES DLPDU.

d) If no such member of the referenced DL-address request queue, Q_A(UR), exists, then

1) If the DL-address is

— a DLCEP-address of a local peer or publisher DLCEP; or

— a publisher DLCEP-address of a local subscriber DLCEP,

and the sending binding of the DLCEP is to a buffer which is not empty, then

i) the DLE shall transmit the DLPDU appropriate to the state of the DLCEP; and

ii) if the resultant transmitted DLPDU completes the transmission of a DLSDU (so that no segments of that DLSDU remain to be transmitted), then the DLE shall notify the DLS-user of the request's completion, if appropriate.

2) Otherwise, when (i) does not apply, and the DL-address is a DLCEP-address of a local peer or publisher DLCEP, then the DLE shall transmit a DT or RC DLPDU, as appropriate for the state of the DLCEP, with a null user-data field, to convey the state of the DLCEP to the correspondent peer or subscriber DLCEP(s).

3) Whether or not a DLPDU was transmitted, the DLE shall advance to the next member of the sequence which was addressed by the received ES DLPDU.

All other cases are erroneous and are not permitted.

7.16.4.1.2.2 Additional considerations

The following considerations also apply:

i) If there are no additional members in the sequence, and if immediate retry of the currently-selected transaction is not possible, then the final-token-use subfield in that non-RT DLPDU can be set to the value FINAL before transmission, thereby returning the delegated token to the LAS DLE at the end of the current transaction.

ii) If the sequence is completed, and the delegated token was not returned by setting a final-token-use subfield to the value FINAL in the last DLPDU transmitted by the DLE, then the DLE shall return the delegated token by transmitting an RT DLPDU.

iii) If at any time

A) the remaining-duration down-counter, C(RD), is inadequate to permit any further use by the receiving DLE; and

NOTE — 5.7.1.11 specifies the interpretation of this constraint when C(RD) is not implemented as a down-counter.

B) the sequence contains additional members,

then the receiving DLE shall return the delegated token to the LAS by sending an RI DLPDU specifying the minimum delegation interval for token use required when the token is next returned.

NOTE — An RI DLPDU, which is sent to terminate use of a token delegated by an ES DLPDU, indicates to the receiving LAS that sequence execution was incomplete and that another ES DLPDU, with a final-token-use subfield specifying CONTINUE, should be sent to the same DLSEP-address.

7.16.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.16.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.16.4.4 Additional actions required of the current LAS DLE

If the DLE did not just send the DLPDU, then the DLE shall drop the scheduler token and deactivate its role as LAS; and shall inform local DL-management of the event.

7.17 Return Token (RT) DLPDU

A RETURN TOKEN (RT) DLPDU is used to return the token delegated by a PT or ES DLPDU to the LAS and to indicate that no additional token delegation, to the DL-address of the last PT or ES DLPDU, is required.

NOTE — The delegated token also may be returned by setting the final-token-use designator in any DLPDU to the value FINAL, and then ceasing use of the delegated token after transmitting that DLPDU.

7.17.1 Structure of the RT DLPDU

Table 20 — Structure of RT DLPDUs

frame control
0011 0100

7.17.1.1 The frame control field shall specify the DLPDU's function.

7.17.1.2 The address field shall be null.

7.17.1.3 The parameters field shall be null.

7.17.1.4 The user-data field shall be null.

7.17.2 Content of the RT DLPDU

The frame control field shall be encoded as specified in Table 20.

The address, parameters and user data fields shall be null.

7.17.3 Sending the RT DLPDU

An RT DLPDU results only from receipt of a PT or ES DLPDU. The RT DLPDU shall be sent as specified in 7.15.4 and 7.16.4.

7.17.4 Receiving the RT DLPDU

A received RT DLPDU shall be treated as follows by the receiving DLE:

7.17.4.1 Actions required of all DLEs

none.

7.17.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.17.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.17.4.4 Additional actions required of the current LAS DLE

When the scheduler token is not dominant on the local link, then

a) if the token being returned was delegated by a PT DLPDU, then the DLE shall set the value of the V(DTA)'th member of the remaining-token-holding-time-array, V(RTHA) (see 5.7.5.13) to zero; and

b) the DLE shall assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.18 Request Interval (RI) DLPDU

A REQUEST INTERVAL (RI) DLPDU is used to return a token delegated by a PT or ES DLPDU to the LAS and to indicate the minimal useful delegation interval required for the next delegation.

NOTE — The delegated token also may be returned by setting the final-token-use designator in any DLPDU to the value FINAL, and then ceasing use of the delegated token after transmitting that DLPDU, or by sending an RT DLPDU.

7.18.1 Structure of the RI DLPDU

Table 21 — Structure of RI DLPDUs

frame control	parameters
0010 0000	DD-p

7.18.1.1 The frame control field shall specify the DLPDU's function.

7.18.1.2 The address field shall be null.

7.18.1.3 The parameter field shall be structured and encoded as specified in 8.10.

7.18.1.4 The user-data field shall be null.

7.18.2 Content of the RI DLPDU

The frame control field shall be encoded as specified in Table 21. The address field shall be null.

The parameters field shall specify the minimum useful duration for the next delegation of the current token. The user data field shall be null.

7.18.3 Sending the RI DLPDU

An RI DLPDU results only from receipt of an ES or PT DLPDU. The RI DLPDU shall be sent as specified in 7.15.4 and 7.16.4.

7.18.4 Receiving the RI DLPDU

A received RI DLPDU shall be treated as follows by the receiving DLE:

7.18.4.1 Actions required of all DLEs

none.

7.18.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.18.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.18.4.4 Additional actions required of the current LAS DLE

When the scheduler token is not dominant on the local link, then the DLE shall

a) if the token being returned was delegated by a PT DLPDU, then the DLE shall set the value of the V(DTA)'th member of the remaining-token-holding-time-array, V(RTHA) (see 5.7.5.13), to account for the link capacity just used by the DLE;

b) if the token being returned was delegated by an ES or PT DLPDU, then the DLE shall forward the DD-parameter field of this DLPDU, together with V(DTA), to the LAS DLE's upper-level functions (see 10.2.2) for attempted rescheduling; and

c) the DLE shall assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.19 CLAIM LAS (CL) DLPDU

A CLAIM LAS (CL) DLPDU is used by an LM DLE to initialize the local link, or to recover from a prolonged period of silence on the local link indicative of the failure of the prior LAS.

7.19.1 Structure of the CL DLPDU

Table 22 — Structure of CL DLPDUs

frame control	**source NODE-address**
0000 0001	N

7.19.1.1 The frame control field shall specify

a) the DLPDU's function; and

b) the length, number and type of DLPDU addresses.

7.19.1.2 The address field shall consist of an explicit source NODE DL-address.

7.19.1.3 The parameters field shall be null.

7.19.1.4 The user-data field shall be null.

7.19.2 Content of the CL DLPDU

The frame control field shall be encoded as specified in Table 22. The address field shall be the NODE DL-address of the DLE which has detected the prolonged link inactivity, whose value is the sender's node-id, V(TN). The parameters and user-data fields shall be null.

NOTE — The primary purposes of the NODE DL-address in the CL DLPDU are

a) to permit improved detection of collisions during the LAS claiming procedure; and

b) to identify the DLE which detected inactivity to observing DLEs, which may have utility in real open systems for identifying DLEs with particularly-obstructive classes of PhE or DLE faults, sometimes characterized as "faulty transmitter" or "deaf receiver."

7.19.3 Sending the CL DLPDU

A CL DLPDU shall result from the detection in an LM DLE of a period of node-id slot-times, V(TN) × V(ST) octet-durations, of continuous link inactivity, implying failure of the previous LAS DLE, if any.

Upon such detection, the DLE shall immediately send one CL DLPDU, and shall again monitor the medium for node-id slot-times, V(TN) × V(ST) octet-durations, of continuous link inactivity after sending the first CL DLPDU.

If no activity is again heard (other than potentially a CL DLPDU with the NODE DL-address that was just sent), then the sending LM DLE shall immediately send a second identical CL DLPDU, after which the DLE shall choose a uniformly-distributed random integer in the range zero to three, and shall monitor the medium for that many slot-times, random(0..3) × V(ST) octet-durations. The actual random choice shall be statistically independent of similar choices made by other DLEs.

NOTE — This requirement for statistical independence minimizes the probability of repeatedly-identical choices by identically-constructed real devices.

If no activity is again heard (other than potentially the CL DLPDU with the NODE DL-address that was just sent), then the sending LM DLE shall activate its LAS functions, assume the scheduler token and commence active operation as the LAS.

7.19.4 Receiving the CL DLPDU

A received CL DLPDU shall be treated as follows by the receiving DLE:

7.19.4.1 Actions required of all DLEs

a) The DLPDU shall be reported to the DLE's local DL-management.

b) If the receiving DLE holds a token on the local link, then the receiving DLE shall drop the token at once and shall inform local DL-management of the event.

7.19.4.2 Additional actions required of a Link-Master class or Bridge class DLE

7.19.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.19.4.4 Additional actions required of the current LAS DLE

The DLE shall drop the scheduler token and deactivate its role as LAS; and shall inform local DL-management of the event.

7.20 Transfer LAS (TL) DLPDU

A TRANSFER LAS (TL) DLPDU is used by the current LAS DLE to transfer the scheduler token and the role of LAS to another LM DLE on the local link. The TL DLPDU is sent only after having been requested by the addressed LM DLE, and may be rejected if the addressed DLE determines that its own copy of the local link's schedule is not current.

7.20.1 Structure of the TL DLPDU

Table 23 — Structure of TL DLPDUs

frame control	**destination NODE-address**	**user data**
0000 0110	N	o-SPDU

7.20.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is URGENT; and

c) the length, number and type of DLPDU addresses.

7.20.1.2 The address field shall consist of an explicit destination NODE DL-address.

7.20.1.3 The parameters field shall be null.

7.20.1.4 The user-data field shall consist of an optional higher-level LAS-database-status SPDU of 64 octets or less.

7.20.2 Content of the TL DLPDU

The frame control field shall be encoded as specified in Table 23. The address field shall be the NODE DL-address of the DLE to which the LAS role is being transferred. The parameters field shall be null.

The user data field shall convey a higher-level LAS-data-base-status SPDU (see Annex B/B.3.2.3) whose maximum size is limited as specified in 7.20.1.4, or shall be null. When present, the contents of this SPDU shall reflect the most recent schedule construction activities, if any, prior to sending the TL DLPDU.

7.20.3 Sending the TL DLPDU

The LAS DLE shall send a TL DLPDU in response to the receipt of an explicit scheduling request received by the LAS from another LM DLE on the local link, requesting transfer of the scheduling-token and the associated LAS role. It is a protocol error to send a TL DLPDU to a non-requesting DLE.

NOTE — Such a request can only be made through a higher-level SPDU sent to the current LAS' DL-support functions. DL-management is not permitted to command such a transfer, though it may be able to prompt the requesting LM DLE to request the transfer directly.

A TL DLPDU may be sent on the link when

a) the sending DLE, the LAS, holds a scheduler token which is the dominant token on the local link;

b) the LAS DLE has completed the last cycle of “circulating the token,” and would otherwise initiate the next cycle of “circulating the token” by explicitly sending a PT DLPDU to the lowest-numbered NODE DL-address represented in V(LL); and

NOTE — This restriction on transferring the role of LAS eliminates the need to transfer (reliably) the current value of the array V(RTHA) just before sending the TL DLPDU.

c) the remaining allocated duration of token usage before the next scheduled activity permits the LAS DLE to repeatedly send a TL DLPDU and monitor the local link for a period of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations, for any immediate response from the addressed DLE.

After sending a TL DLPDU, the LAS DLE shall monitor the local link for a period of immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations, waiting for a response:

1) If a PhL-indication (see 5.4.3) reporting DATA is received, then the DLE shall

— stop further monitoring;

— then wait for receipt of the PhL-indication reporting END-OF-DATA-AND-ACTIVITY or END-OF-ACTIVITY; and

— then proceed as in (3).

2) If (1) does not apply, and the monitoring period expires, then

i) if the bus is not active at that moment (that is, the last-received PhL-indication reported END-OF-ACTIVITY), then the DLE shall proceed as in (4).

ii) if (i) does not apply, implying that the link is still active at that moment (that is, the last-received PhL-indication reported START-OF-ACTIVITY), then

a) Unless (*b*) applies, the DLE shall monitor the local link for a period of one additional slot-time, V(ST) octet-durations, waiting for a PhL-indication:

A) If a PhL-indication reporting DATA is received, then the DLE shall proceed as in (1).

B) If a PhL-indication reporting END-OF-ACTIVITY is received, and (A) does not apply, then the DLE shall stop further monitoring and proceed as in (4).

C) If neither (A) nor (B) applies, and the monitoring period expires before a PhL-indication is received, then the DLE shall

—wait for receipt of the PhL-indication reporting END-OF-DATA-AND-ACTIVITY or END-OF-ACTIVITY; and
—then proceed as in (3).

b) Implementations based on hardware that was designed specifically to implement this International Standard, where such hardware was demonstrable on or before 31 December 1995 alternatively may just proceed as in (*a*.C) without any additional period of monitoring.

NOTE — After this document achieved initial ACDV status, implementors were encouraged to develop chips to assist in evaluating this complex protocol. As a consequence, it was found desirable to improve the noise rejection characteristics of the token passing process, and the text (2.ii.*a*), which would require changes in those existing chips, was the result. (*b*) grandfathers the noise rejection approach of the first ACDV for those early implementations, and only for those early implementations.

3) If the link activity was sufficient to infer that DATA was received as in (1), (2.ii.*a*.A) or (2.ii.*a*.C), then

i) If the link activity did not result in a DLPDU, then the LAS DLE shall drop its scheduler token and shall continue or resume monitoring the local link until either (3.i.A) or (3.i.B) occurs.

NOTE: This monitoring provides for recovery of the LAS role when the transfer is unsuccessful.

A) If the monitoring DLE observes a DLPDU (with correct FCS) then the monitoring DLE shall cease monitoring for LAS role recovery, and shall inform local DL-management of the event.

B) If the monitoring DLE observes a period of fifteen slot-times, 15 × V(ST) octet-durations, of continuous non-activity on the local link, then the monitoring DLE shall

— re-assume the scheduler token;

— inform local DL-management of the event; and

— resume active operation as the LAS, and commence transmission on the link.

ii) If the link activity resulted in a DLPDU (with correct FCS) then

A) if the received DLPDU is an SR DLPDU, specifying a reason of "failure — LAS transfer rejected," then the monitoring DLE shall

— re-assume the scheduler token;

— inform local DL-management of the event; and

— resume active operation as the LAS, and commence transmission on the link.

B) else if the received DLPDU is any other DLPDU, then the monitoring DLE shall cease monitoring for LAS role recovery, and shall inform local DL-management of the event.

4) If there was no link activity, or the link activity was not sufficient to infer that DATA was received as in (2.i) or (2.ii.*a*.B), then the LAS DLE may retry re-sending the TL DLPDU, up to the number of times specified as the maximum-retry-count, V(MRC) (see 5.7.1.5), of the local link.

If all attempted retries are unsuccessful, then the LAS DLE shall

— retain the scheduler token and continue active operation as the LAS;

— inform local DL-management of the event; and

— start the next transmission within token-recovery-delay, P(TRD) × V(ST) octet-durations, of the beginning of the current period of link non-activity.

7.20.4 Receiving the TL DLPDU

A received TL DLPDU shall be treated as follows by the receiving DLE:

7.20.4.1 Actions required of all DLEs

The DLPDU shall be reported to the DLE's local DL-management.

7.20.4.2 Additional actions required of a Link-Master class or Bridge class DLE

If the destination DL-address specified by the DLPDU designates the DLE's NODE DL-address, then

a) If

1) the receiving DLE is not awaiting receipt of the TL DLPDU;

2) the receiving DLE cannot execute the existing schedule, either due to schedule complexity or length, or because the receiving DLE cannot meet the schedule overhead constraint, V(MSO) (see 5.7.5.6), built into the existing schedule; or

3) if the schedule construction and live-list information conveyed in the LAS-database-status SPDU within the received TL DLPDU indicates that the receiving DLE does not have a current copy of either the schedule, or the live-list, or both,

then the receiving DLE shall reply with an SR DLPDU within a period of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, as measured at the receiving DLE, with a status of "failure —LAS transfer rejected."

b) Otherwise, when a) does not apply, then the receiving DLE shall assume the scheduler token, activate its LAS functions, and re-commence operation as the LAS.

7.20.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.20.4.4 Additional actions required of the current LAS DLE

If the DLE did not just send the DLPDU, then the DLE shall drop the scheduler token and deactivate its role as LAS; and shall inform local DL-management of the event.

7.21 Wakeup (WK) DLPDU

A WAKEUP (WK) DLPDU is used by the current LAS DLE to stimulate a class B or class C fractional duty cycle (FDC) DLE on the local link (see 3.3.32) to transition to a fully-operational state, capable of direct communication.

Transmission of the WK DLPDU usually occurs as a result of previous explicit scheduling, either by the addressed FDC DLE or by DL-management. However, the LAS DLE, which must also be a bridge DLE to support FDC DLEs, may transmit the WK DLPDU on its own initiative, typically when its bridge-function's queue of DLPDUs waiting for transmission to the FDC DLE exceeds some internal threshold.

7.21.1 Structure of the WK DLPDU

Table 24 — Structure of WK DLPDUs

frame control	destination NODE-address
0000 0000	N

7.21.1.1 The frame control field shall specify

a) the DLPDU's function;

b) the DLPDU's implicit priority, which is URGENT; and

c) the length, number and type of DLPDU addresses.

7.21.1.2 The address field shall consist of an explicit destination NODE DL-address.

7.21.1.3 The parameters field shall be null.

7.21.1.4 The user-data field shall be null.

7.21.2 Content of the WK DLPDU

The frame control field shall be encoded as specified in Table 24. The address field shall be the NODE DL-address of the DLE which is to receive the WK DLPDU. The parameters and user-data fields shall be null.

NOTE — The specific encoding of the WK DLPDU minimizes the amount of active DLPDU-recognition circuitry required within a “sleeping” FDC DLE.

7.21.3 Sending the WK DLPDU

The LAS DLE shall send a WK DLPDU when requested by prior explicit scheduling. The link capacity used for sending the WK DLPDU shall be deducted (to the extent possible) from the remaining allocated duration of scheduler token usage for scheduled traffic, which is V(MST) × V(TTRT) octet-durations, within the current cycle of “circulating the token.”

A bridge DLE may also send a WK DLPDU when so requested by the DLE’s own bridge-forwarding functions when the DLE holds a scheduler token or delegated token which is the dominant token on the local link, and when the remaining allocated duration of token usage prior to the next scheduled activity permits completion of the DLPDU’s transmission prior to expiration of the token.

NOTE — Means of ensuring that the FDC DLE’s duty cycle requirements are not exceeded due to bridge-forwarding-induced over-scheduling is a matter of current study.

7.21.4 Receiving the WK DLPDU

A received WK DLPDU shall be treated as follows by the receiving DLE:

7.21.4.1 Actions required of all DLEs

If the destination DL-address specified by the DLPDU designates the DLE’s NODE DL-address, then the DLE shall activate the remainder of its DLE functions and prepare for active communication on the local link.

NOTE — Receipt of this DLPDU has no impact on non-FDC DLEs.

7.21.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.21.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.21.4.4 Additional actions required of the current LAS DLE

none.

7.22 IDLE (IDLE) DLPDU

An IDLE DLPDU is used by the current token holder to place a DLPDU on the local link which is ignored by all DLEs, other than for its impact on each DLE’s sense of link activity, but which may convey information to, or trigger, a data link analysis device.

7.22.1 Structure of the IDLE DLPDU

Table 25 — Structure of IDLE DLPDUs

frame control	user data
0001 0F10	o-DLSDU

7.22.1.1 The frame control field shall specify

a) the DLPDU's function; and

b) whether or not transmission of the current DLPDU terminates use of a delegated token.

7.22.1.2 The address field shall be null.

7.22.1.3 The parameters field shall be null.

7.22.1.4 The data field shall consist of a single DLSDU whose maximum size is the maximum permitted for an URGENT priority DLPDU. This data field may be null.

7.22.2 Content of the IDLE DLPDU

The frame control field shall be encoded as specified in Table 25. The address and parameter fields shall be null.

The contents of the user-data field, possibly null, is not specified by this standard.

7.22.3 Sending the IDLE DLPDU

An IDLE DLPDU may be sent on the link when the sending DLE holds a scheduler token or delegated token which is the dominant token on the local link, and when the remaining allocated duration of token usage, C(RD), permits completion of the IDLE DLPDU's transmission prior to expiration of the token.

NOTE — The LAS DLE uses the IDLE DLPDU to fill otherwise-unused link capacity until the time of the next scheduled link activity, thereby preventing other DLEs from inferring the LAS DLE's failure from the lack of activity on the link. To facilitate this use, the IDLE DLPDU has a minimum size of one octet.

A local management request may result in the transmission of an IDLE DLPDU.

If the DLE holds a delegated token, and this DLPDU is the final DLPDU which needs to be sent on this use of the delegated token, then the DLE may set the final-token-use subfield of the IDLE DLPDU to the value FINAL; else that subfield shall have the value NOT-FINAL.

7.22.4 Receiving the IDLE DLPDU

A received IDLE DLPDU shall be treated as follows by the receiving DLE:

7.22.4.1 Actions required of all DLEs

The DLE shall ignore the DLPDU on receipt, except for possible local management reporting or special instrumentation purposes.

7.22.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.22.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.22.4.4 Additional actions required of the current LAS DLE

If the final-token-use subfield of the received DLPDU has the value FINAL, then the LAS DLE shall

a) assume that the current use of the delegated token is terminated as if the token had been returned by an RT (see 7.17) DLPDU; and

b) assume that the scheduler token is again dominant on the local link and resume active operation as the LAS.

7.23 Spare DLPDUs

A SPARE DLPDU is reserved for future standard use, and may not be sent by any DLE.

7.23.1 SPARE DLPDU code points

Table 26 — Assumed structure of undefined (spare) DLPDUs

format	frame control
1	0000 0x11
2	0010 0x01
3	00xx 1xxx
4	0100 xxxx
5	101x xx00

A SPARE DLPDU shall consist of a frame control field. Each spare FC code point is reserved for future standard use.

The frame format and contents of spare code points are subject to change if and when used in a revision of this part of this International Standard. Until that time they shall not be used.

7.23.1.1 The frame control field shall specify the DLPDU's function — reserved spare. No DLPDU priority is implied.

7.23.1.2 The address field shall be null.

7.23.1.3 The parameters field shall be null.

7.23.1.4 The user-data field shall be null.

7.23.2 Content of a SPARE DLPDU

The frame control field shall be encoded as specified in Table 26. The address, parameters and user-data fields shall be null.

7.23.3 Sending a SPARE DLPDU

SPARE DLPDUs may not be sent or forwarded.

7.23.4 Receiving a SPARE DLPDU

A received SPARE DLPDU shall be treated as follows by the receiving DLE:

7.23.4.1 Actions required of all DLEs

The DLE shall ignore the DLPDU on receipt, except for possible local management reporting or special instrumentation purposes.

7.23.4.2 Additional actions required of a Link-Master class or Bridge class DLE

none.

7.23.4.3 Additional actions required of a Bridge class DLE

The DLE shall not forward the DLPDU.

7.23.4.4 Additional actions required of the current LAS DLE

none.

7.24 Reserved (can't use) DLPDUs

A RESERVED (CAN'T USE) DLPDU is reserved for future standard use, and may not be sent by any DLE.

7.24.1 RESERVED (CAN'T USE) DLPDU code points

Table 27 — Assumed structure of RESERVED (CAN'T USE) DLPDUs

format	**frame control**
1	0000 0010
2	0000 010x
3	0010 001x
4	0010 0100
5	0011 0000
6	0101 1x00
7	1001 1x00
8	110x 1x00

The FC code point for a RESERVED (CAN'T USE) DLPDU shall not be used; these code points are reserved to simplify decoding of the FC octets of other defined DLPDUs. The set of such reserved code points is shown in Table 27.

The frame control field shall specify the DLPDUs function — reserved (can't use). It shall not be used.

NOTE — Since these FC code points are reserved to simplify DLE FC-code-point decoding, their recognition as being distinct from structurally-related FC code points cannot be mandated.

7.24.2 Content of a RESERVED (CAN'T USE) DLPDU

No content is defined, since DLPDUs with these code point may not be originated.

7.24.3 Sending a RESERVED (CAN'T USE) DLPDU

DLPDUs specifying RESERVED (CAN'T USE) FC code points shall not be originated. However, they may be forwarded by a bridge if they were not discarded upon receipt.

7.24.4 Receiving a RESERVED (CAN'T USE) DLPDU

If the receiving DLE distinguishes between RESERVED (CAN'T USE) and other DLPDUs, then the DLE should ignore the RESERVED (CAN'T USE) DLPDU on receipt (except for possible local management reporting or special instrumentation purposes). However, the ability to make such a distinction and to perform such discarding is not required.

8 DLPDU-parameter structure and encoding

Boolean variables are all encoded with a common representation:

0: FALSE

1: true

The 3-bit DL protocol version number (VVV), which is used in a number of the DLPDU parameters and SPDUs (see Annex B), shall have the value one (1).

8.1 Structure and encoding of EC-PARAMETERS

An EC DLPDU is used to establish a peer DLC between two DLS-users, or a multi-peer DLC between a publishing DLS-user and subscribing DLS-users.

Table 28 — Structure of an EC DLPDU's parameters

DLC basic attributes	DLC attributes when sending	DLC attributes when receiving

The connection parameters (EC-parameters) field, which partitions roughly as shown in Table 28, shall specify the parameters for the proposed DLC:

a) The first two octets, ordered as shown in Tables 29 and 30, specify basic attributes of the DLC, beginning with the DL protocol version number:

Table 29 — EC-parameters: 1st octet

Reply Request	**Publisher-DLCEP-address Reuse Discriminator**			**Path Diversity**	**DL Protocol Version Number**		
R	NNN			Q	VVV		
7	6	5	4	3	2	1	0

1) a 1-bit reply-request subfield (R), specifying whether a reply is requested (=1) or not (=0);

NOTE — This field is provided to ensure proper action independent of any state of the receiving DLCEP.

2) a 3-bit publisher-DLCEP-address reuse-discriminator subfield (NNN), designating the current set of EC-parameters associated with a PUBLISHER DLCEP-address

i) encoded as zero, when the sender's DLCEP-class (CC) is not PUBLISHER; and

ii) assigned by the DLE, when the sender's DLCEP-class (CC) is PUBLISHER

NOTE — The DLE assigns a value to this subfield whenever it receives a request to initiate a new PUBLISHER DLCEP from either a local DLS-user in a DL-CONNECT request primitive, or from a remote DLS-user in an EC DLPDU. The choice of value can be random, or can be based on knowledge of recently-used values which the DLE avoids.

This subfield is used to discriminate between two different DLCs using the same publisher-DLCEP-address. This could happen when the first DLC had been disconnected and the publishing DLE used the same DLCEP-address for establishing the second DLC, which is different than the first DLC. If a subscriber to the first DLC did not receive the DC DLPDU for that DLC, it could use this field of the just received EC DLPDU to determine that the received EC DLPDU is for a different DLC. Therefore, if the publisher-DLCEP should send an EC DLPDU for the same DLC, it does not change this subfield.

3) a one-bit DLL path-diversity subfield (Q), encoded as:

0: ANY-PATH;

1: THIS-PATH — use the path on which this EC DLPDU was received;

NOTE — This subfield provides a means by which a DLE can restrict all communications at a DLCEP to a specific DL-path, thereby providing a means for testing the specific DL-path. The means by which the THIS-PATH value is selected, and by which the actual path employed is chosen, is a DLE-local issue.

4) a 3-bit DL protocol version number (VVV), whose value is specified in 8;

Table 30 — EC-parameters: 2nd octet

sender's DLCEP class		**DLL Priority**		DLL Address **Size**		DLPDU authentication	
CC		PP		SS		XX	
7	6	5	4	3	2	1	0

5) a two-bit sender's DLCEP-class subfield (CC), encoded as:

00: reserved for future standard use;

01: PEER DLCEP;

10: PUBLISHER DLCEP;

11: SUBSCRIBER DLCEP;

6) a two-bit DLL priority subfield (PP), encoding the DLL priority as (see 6.2.1.3):

00: reserved for future standard use;

01: URGENT (high) priority;

10: NORMAL (medium) priority;

11: TIME-AVAILABLE (low) priority;

7) a two-bit DLL address size subfield (SS), encoding the proposed DLL address size (see 6.2.1.1) as:

00: VERY-SHORT — addresses are omitted where possible;

01: SHORT;

10: LONG;

11: reserved for future standard use;

8) a two-bit DLPDU-authentication subfield (XX), encoding the required DLPDU authentication (see 5.2.4) as:

00: ORDINARY;

01: reserved for future standard use;

10: SOURCE;

11: MAXIMAL;

b) The next four octets, ordered as shown in Tables 31 and 32, specify the maximum confirm delay, in units of 1 ms, for the sender's DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY request primitives, and DL-DATA request primitives, respectively. The DLS-user value UNLIMITED shall be encoded as the value $FFFF_{16}$ (all ones).

Table 31 — EC-parameters: 3rd and 4th octets

Sender's Maximum Confirm Delay for DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY requests															
MCD_CRS															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

and

Table 32 — EC-parameters: 5th and 6th octets

Sender's Maximum Confirm Delay for DL-DATA requests															
MCD_D															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOTE —Each two-octet maximum confirm delay is sent most-significant-octet first.

c) The next four octets, ordered as shown in Tables 33, 34 and 35, specify proposed attributes for the sender's sending direction of data flow:

Table 33 — EC-parameters: 7th octet

Sender's Sending DLC Attributes								
DLCEP Data Delivery Features		**Residual Activity**	Queue / Buffer	**Maximum Window Size**				
TT_S		A_S	B_S	$WWWW_S$				
7	6	5	4	3	2	1	0	

1) a two-bit sending DLCEP data-delivery features subfield (TT_S), encoded as:

00: UNORDERED DLC;

01: ORDERED DLC;

10: DISORDERED DLC;

11: CLASSICAL DLC;

2) a one-bit residual activity subfield (A_S), indicating whether residual activity in the specified (sender-to-receiver) direction will be provided when there are no unacknowledged DLSDUs, encoded as a Boolean;

NOTE — Residual activity is required when MAXIMAL DLPDU-authentication is specified, or upon DLS-user or DL-management direction. Other methods of requesting residual activity are for future study.

3) a one-bit buffer/queue subfield (B_S), indicating whether the source in the specified (sender-to-receiver) direction is a buffer (=1) or a queue (=0);

4) a four-bit window-size subfield ($WWWW_S$), indicating the maximum number of previously-sent DLSDUs which will be available for retransmission, and encoded as zero when transmission in that direction is not permitted on the DLC;

Table 34 — EC-parameters: 8th octet

Sender's Sending DLC Attributes (cont.)							
basic-DLC-parameters format			**2-way data exchange**	**reserved**	**Timeliness included**	**Time stamp format**	
FFF_S			E_S	0	G_S	HH_S	
7	6	5	4	3	2	1	0

5) a three-bit format subfield (FFF_S), giving the desired format of the basic-DLC-parameters portion of the Status-Data-parameters (SD-parameters) in the specified direction, encoded as a three-bit integer:

0: format A—[8.4.2(A)];

1: format B—[8.4.2(B)];

2: format C—[8.4.2(C)];

3: reserved for future standard use;

4: format D—[8.4.2(D)];

5: format E—[8.4.2(E)];

6: format F—[8.4.2(F)];

7: format G—[8.4.2(G)].

6) a one-bit 2-way data exchange subfield (E_S), indicating whether DLSDU data may be sent in the sender-to-receiver direction by ED DLPDUs, encoded as a Boolean:

7) a one-bit subfield reserved for future standards use, encoded as zero;

8) a one-bit timeliness-included subfield (G_S), indicating the presence of a timeliness parameter (see 8.4.2.1(b.3) and 8.4.2.1(c.5)) associated with transmitted DLSDUs in transmitted DLPDUs, encoded as a Boolean;

9) a two-bit time-stamp-format subfield (HH_S), indicating whether a DL-time-stamp (see 8.4.2.2) is included in SD-parameters which accompany transmitted DLSDUs, encoded as:

00: format J—[8.4.2(J)], which is a null field with no time stamp;

01: format K—[8.4.2(K)], which is a two-octet field with a time stamp period of over 2 s;

10: format L—[8.4.2(L)], which is a three-octet field with a time stamp period of over 8 min;

11: format M—[8.4.2(M)], which is a six-octet field with a time stamp period of over 100 yr;

Table 35 — EC-parameters: 9th and 10th octets

Sender's Sending DLC Attributes (cont.)															
Maximum DLSDU Size															
$M \ldots M_S$															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

10) a two-octet subfield ($M \ldots M_S$), specifying the maximum size DLSDU that can be sent on the DLC.

NOTES

1. The two-octet maximum size is transmitted most-significant-octet first.

2. The range for each parameter is dependent on the DLC's priority, and can range between zero and 16 times the maximum data length specified in 6.2.4 for the corresponding priority, inclusively.

d) The next four octets, identical in form to those specified in (c) and ordered as shown in Tables 36 through 38, specify proposed attributes for the sender's receiving direction of data flow:

Table 36 — EC-parameters: 11th octet

Sender's Receiving DLC Attributes							
DLCEP Data Delivery Features		**Residual Activity**	**Queue / Buffer**	**Maximum Window Size**			
TT_R		A_R	B_R	$WWWW_R$			
7	6	5	4	3	2	1	0

1) a two-bit receiving DLCEP data delivery features subfield (TT_R), encoded as specified in (c.1);

2) a one-bit residual activity subfield (A_R), indicating whether residual activity in the specified (receiver-to-sender) direction is required, encoded as specified in (c.2);

NOTE — See NOTE after (c.2).

3) a one-bit buffer/queue subfield (B_R), indicating whether the sink in the specified (receiver-to-sender) direction is a buffer or a queue, encoded as specified in (c.3);

4) a four-bit window-size subfield ($WWWW_R$), indicating the maximum number of unacknowledged DLSDUs which can be usefully received;

Table 37 — EC-parameters: 12th octet

Sender's Receiving DLC Attributes (cont.)							
Basic-DLC-parameters format			**2-way data exchange**	**Reserved**	**Timeliness included**	**Time stamp format**	
FFF_R			E_R	0	G_R	HH_R	
7	6	5	4	3	2	1	0

5) a three-bit format subfield (FFF_R), giving the desired format of basic-DLC-parameters portion of the SD-parameters in the specified direction, encoded as specified in (c.5).

6) a one-bit 2-way data exchange subfield (E_R), indicating whether DLSDU data may be sent in the receiver-to-sender direction by ED DLPDUs, encoded as a Boolean;

7) a one-bit subfield reserved for future standards use, encoded as zero;

8) a one-bit timeliness-included subfield (G_R), indicating the presence in received DLPDUs of a timeliness parameter (see 8.4.2.1(b.3) and 8.4.2.1(c.5)) associated with received DLSDUs, encoded as a Boolean;

9) a two-bit time-stamp-format subfield (HH_R), indicating whether a DL-time-stamp is included in SD-parameters which accompany received DLSDUs, encoded as specified in (c.9);

Table 38 — EC-parameters: 13th and 14th octets

Sender's Receiving DLC Attributes (cont.)															
Maximum DLSDU Size															
$M \ldots M_R$															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

10) a two-octet subfield ($M \ldots M_R$), specifying the maximum size DLSDU that can be received on the DLC.

NOTES — see NOTES following (c.10).

8.2 Structure and encoding of DC-PARAMETERS

A DISCONNECT CONNECTION (DC) DLPDU is used to disconnect an existing or proposed DLC.

The Disconnect-Connection-parameters (DC-parameters) field shall specify the desired DLC-support action and reason, together with any associated operational parameters. This parameter field is two octets, ordered as shown in Tables 39, 40 and 41.

Table 39 — DC-parameters and RC-parameters: 1st octet

Reply Request	**reserved**				**DL Protocol Version Number**		
R	0000				VVV		
7	6	5	4	3	2	1	0

a) a one-octet subfield, coded as in Table 39, consisting of

1) a one-bit subfield (R) specifying whether a reply is requested (=1) or not (=0);

NOTE — This field is provided to ensure proper action independent of any state of the receiving DLCEP.

2) a four-bit subfield reserved for future standards use and encoded as zero; and

3) a 3-bit DL protocol version number (VVV), whose value is specified in 8; and

Table 40 — DC-parameters and RC-parameters: 2nd octet

Reason for DLPDU							
Reason							
7	6	5	4	3	2	1	0

b) a one-octet subfield, specifying the reason for the requested DLC support action, based on 18.2.3 and 18.2.4 of Part 3 of this International Standard, and coded in hexadecimal as specified in Table 41. All unused reason codes in Table 41 in the hexadecimal range 40 through 7F are reserved for future standards use; the other codes in the range 00 through 3F may be used as desired, and can be interpreted as reason unspecified (by this part of this International Standard).

8.3 Structure and encoding of RC-PARAMETERS

A RESET CONNECTION (RC) DLPDU is used to reset or disconnect an existing or proposed DLC.

The Reset-Connection-parameters(RC-parameters) field shall specify the desired DLC-support action and reason, together with any associated operational parameters. This parameter field is four octets, ordered as shown in Tables 39, 40, 42, 43 and 43b:

a) A one-octet subfield, coded as specified in 8.2(a).

Table 41 — Disconnect reasons

code	Reason for disconnect	Reason Class
00	user-originated disconnection — normal condition	user-originated disconnection (00 .. 1F)
02	user-originated disconnection — abnormal condition	
1E	user-originated disconnection or connection rejection — reason unspecified	
20	user-originated connection rejection — connection not authorized, permanent condition	user-originated connection rejection (20 .. 3F)
21	user-originated connection rejection — unacceptable QoS, permanent condition	
22	user-originated connection rejection — non-QoS reason, permanent condition	
24	user-originated connection rejection — transient condition	
40	provider-originated disconnection — incorrect DLCEP pairing, permanent condition	provider-originated disconnection (40 .. 5F)
41	provider-originated disconnection — wrong publisher-DLCEP-address reuse-discriminator, permanent condition	
42	provider-originated disconnection — other permanent condition	
43	provider-originated disconnection — wrong DLPDU format or parameters, permanent condition	
44	provider-originated disconnection — wrong DLSDU size, permanent condition	
45	provider-originated disconnection — transient condition	
46	provider-originated disconnection — timeout	
5E	provider-originated disconnection or connection rejection — reason unspecified	
60	provider-originated connection rejection — DL(SAP) address unknown	provider-originated connection rejection (60 .. 7D)
62	provider-originated connection rejection — DLSAP unreachable, permanent condition	
64	provider-originated connection rejection — DLSAP unreachable, transient condition	
65	provider-originated connection rejection — inconsistent DLCEP state, permanent condition	
66	provider-originated connection rejection — QoS unavailable, permanent condition	
68	provider-originated connection rejection — QoS unavailable, transient condition	
7E	disconnection or connection rejection, unknown origin — reason unspecified	unknown origin or type (7E .. 7F)

b) A one-octet subfield, specifying the reason for the requested DLC support action, based on 19.3.2.3 and 19.3.2.4 of Part 3 of this International Standard, and coded in hexadecimal as specified in Table 42. All unused reason codes in Table 42 in the hexadecimal range C0 through FF are reserved for future standards use; the other codes in the range 80 through BF may be used as desired, and can be interpreted as reason unspecified (by this part of this International Standard).

Table 42 — Reset reasons

code	Reason for reset	Reason Class
80	user-originated reset — resynchronization after user timeout	user-originated reset (80 .. BF)
82	user-originated reset — resynchronization after user-detected user-state inconsistencies	
9E	user-originated reset — reason unspecified	
C0	provider-originated reset — resynchronization after activation of a DL-management-established DLCEP	provider-originated reset (C0 .. FD)
C2	provider-originated reset — resynchronization after timeout	
C4	provider-originated reset — resynchronization after maximum number of retransmission requests or attempts	
C6	provider-originated reset — resynchronization after detected sequence number error	
C8	provider-originated reset — resynchronization after other detected DLCEP state inconsistencies	
FC	provider-originated reset — reason unspecified	
FE	reset, unknown origin — reason unspecified	unknown origin (FE .. FF)

Table 43 — RC-parameters: 3rd octet

Modulus Number Preceding Next DLSDU to be Sent, if any							
any value			NDS mod 2^5				
7	6	5	4	3	2	1	0

c) A one-octet subfield, specifying in its low-order bits the low-order five bits of the sequence number NDS preceding the sequence number of the next DLSDU to be sent. The high-order three bits of this octet may have any value (for example, the next higher-order bits of NDS) and shall be ignored on reception (see Table 43).

Table 43b — RC-parameters: 4th octet

Modulus Number of Last Complete DLSDU Received, if any							
any value			NDR mod 2^5				
7	6	5	4	3	2	1	0

d) A one-octet subfield, specifying in its low-order bits the low-order five bits of the sequence number NDR of the last DLSDU received, if any. The high-order three bits of this octet may have any value (for example, the next higher-order bits of NDR) and shall be ignored on reception (see Table 43b).

8.4 Structure and encoding of SD-Parameters

8.4.1 SD-Parameters in DLPDUs addressed to a DL(SAP)-address

A COMPEL ACKNOWLEDGMENT (CA) DLPDU addressed to a DL(SAP)-address is used as the first phase of the remotely-confirmed unitdata transfer service to transfer (or retransfer) a limited amount of transparent user data from one DLS-user to another DLS-user and to request the status of the transfer from the receiving DLE.

A COMPEL DATA (CD) DLPDU addressed to a DL(SAP)-address is used as the first phase of the unitdata exchange service to request the immediate transfer (or retransfer) of a limited amount of transparent user data from one DLS-user to another DLS-user. A CD DLPDU always contains a non-null SD-parameters field.

An EXCHANGE DATA (ED) DLPDU addressed to a DL(SAP)-address is used as the first phase of the unitdata exchange service to transfer (or retransfer) a limited amount of transparent user data from one DLS-user to another DLS-user, and to request the immediate transfer (or retransfer) of a limited amount of transparent user data from that second DLS-user to the first DLS-user. An ED DLPDU always contains a non-null SD-parameters field.

A DATA (DT) DLPDU addressed to a DL(SAP)-address is used

1) to transfer a limited amount of transparent user data from one requesting DLS-user to one or more other DLS-users without establishing or later releasing a DLC; or

2) as the second phase of the unitdata exchange service or remotely-confirmed unitdata-transfer service to acknowledge the transfer of such data without establishing or later releasing a DLC.

The DLPDUs of (1) always contain a null SD-parameters field; the DLPDUs of (2) always contain a non-null SD-parameters field.

NOTE — These DLPDUs are also used within the context of a DLC to provide DL-DATA services.

The various SD-parameter formats, and their potential applicability to the CA, CD, DT and ED DLPDUs addressed implicitly or explicitly to a DLSAP-address, shown in Table 44, are:

Table 44 — Structure of connectionless-mode CA, CD, DT and ED DLPDUs

Connectionless-mode DLPDUs					
format	**frame control**	**d-DLSAP-address**	**s-DLSAP-address**	**parameters**	**user data**
1	CA	[HL.]N.S	[HL.]N.S	format R	DLSDU
2	CD	[HL.]N.S	[HL.]N.S	format R	—
3	ED	[HL.]N.S	[HL.]N.S	format R	DLSDU
4	DT 1	[HL.]N.S	[HL.]N.S	format P	DLSDU
5	DT 2	[HL.]N.S	—	format U	o-DLSDU
6	DT 3	implied PSA	—	format U	—

format	**SD-parameters** / **non-DLC-parameters**
P	— (null) —
R	
U	

a) The **null** format, format P, of length zero octets, is used for the unitdata transfer service. It is always used in DT DLPDUs which have two explicit addresses (see 7.7.1 formats 1L and 1S).

b) The **initiator** format, format R (Table 45), of length one octet, is used for the first phase of the two-phase transaction required for unitdata-exchange and remotely-confirmed unitdata-transfer services. It is always used in CA, CD and ED DLPDUs, all of which always have two explicit addresses (see 7.4.1, 7.5.1 and 7.6.1 formats 1L and 1S).

Table 45 — Short format SD-parameters for connectionless transaction initiators

reserved		**DLSDU priority and presence**		**initiator's transaction index**			
ZZ		PP		N(LTI)			
7	6	5	4	3	2	1	0

1) a two-bit reserved subfield, reserved for future standards use, which shall be zero;

2) a two-bit DLSDU-priority-and-presence subfield, specifying the presence and DLL priority of the accompanying DLSDU (see 6.2.1.3), encoded as:

00: no accompanying DLSDU;

01: URGENT (high) priority accompanying DLSDU;

10: NORMAL (medium) priority accompanying DLSDU;

11: TIME-AVAILABLE (low) priority accompanying DLSDU;

3) a four-bit transaction-index subfield, specifying the transaction index assigned during CA, CD or ED formation and transmission as specified in 7.4.3, 7.5.3 and 7.6.3.

c) The **responder** format, format U (Table 46), of length one octet, is used for the second phase of the two-phase transaction required for unitdata-exchange and remotely-confirmed unitdata-transfer services. It is always used in DT DLPDUs which do not have an explicit source address (see 7.7.1 formats 2L, 2S and 4):

Table 46 — Short format SD-parameters for connectionless responders

status				**initiator's transaction index**			
XXXX				N(LTI)			
7	6	5	4	3	2	1	0

1) a four-bit status subfield, encoded as specified in Table 47; and

2) a four-bit transaction-index subfield, identical to (B.3).

8.4.2 SD-Parameters in DLPDUs addressed to a DLCEP

A COMPEL ACKNOWLEDGMENT (CA) or EXCHANGE DATA (ED) DLPDU addressed to a DLCEP is used to transfer (or retransfer) a limited amount of transparent user data from one DLS-user to one or more other DLS-users and to request the status of the remote DLCEP.

A COMPEL DATA (CD) or EXCHANGE DATA (ED) DLPDU addressed to a DLCEP is used to request the immediate transfer (or retransfer) of a limited amount of transparent user data from one DLS-user to one or more other DLS-users. A CD DLPDU may contain an SD-parameters field; a CA or ED DLPDU always contains an SD-parameters field.

A DATA (DT) DLPDU addressed to a DLCEP is used to transfer a limited amount of transparent user data from one requesting DLS-user to one or more other DLS-users within the context of a DLC; or to acknowledge the transfer of such data within the context of a DLC. These DLPDUs always contain an SD-parameters field (which may be null).

All four types of DLPDU assist in the synchronization of DLCEPs and of their DLS-users:

NOTE — These DLPDUs are also used outside the context of a DLC to provide DL-UNITDATA services.

a) When a CD DLPDU is being sent by an LAS DLE, or such a DLPDU is being forwarded by a bridge, then the DLPDU shall not contain an explicit source DLSAP-address and its parameter field (SD-parameters) shall be null.

Table 47 — Reply status for unitdata-acknowledgment and exchange-unitdata-reply DT DLPDUs

Short Name	Definition	hexadecimal coding
OK	success — no reply data available at responder or service does not provide this information	0
OK_U	success — URGENT priority reply data available at responder	1
OK_N	success — NORMAL priority reply data available at responder	2
OK_TA	success — TIME-AVAILABLE priority reply data available at responder	3
RR	failure — resource limitation in responder — no reply data available at responder or service does not provide this information (note 1)	4
RR_U	failure — resource limitation in responder — URGENT priority reply data available at responder (note 1)	5
RR_N	failure — resource limitation in responder — NORMAL priority reply data available at responder (note 1)	6
RR_TA	failure —resource limitation in responder — TIME-AVAILABLE priority reply data available at responder (note 1)	7
RF	failure —fault in responder	8
RI	failure —responder DL(SAP)-role incompatible with this DLPDU (note 2)	9
RA	failure — response restricted to a different peer DLSAP-address (note 3)	A
—	reserved for compatibility with the SR DLPDU's reply status — not available for other use	B
BF	failure —fault in intermediary bridge	C
BR	failure —resource limitation in intermediary bridge	D
BOK	reserved for interim success — intermediary bridge is forwarding transaction	E
DR	interim success — delayed reply; end station needs more time to prepare response	F

NOTES

1. This status can occur when an addressed responder cannot buffer the received DLSDU (possibly because of a queue-full condition). In such a case the responder is not permitted to send a DLSDU in reply.
2. This status can be generated only when the DL(SAP)-role associated with the received destination DLSAP-address is BASIC or INITIATOR.
3. This status can be generated only when the DL(SAP)-role associated with the received destination DLSAP-address is CONSTRAINED RESPONDER.

b) When a CA, CD, DT or ED DLPDU is being sent from a DLCEP, or being forwarded by a bridge, and the DLPDU does not contain user data, then the negotiated parameter field shall be present, and shall have two sub-parts:

1) the first shall have fixed structure and length [formats A – G] as determined during the DLC establishment process (see 8.1(c.3) or 8.1(d.3)); and

2) the second either

— shall have the fixed structure and length [formats J – M] determined during the DLC establishment process (see 8.1(c.3) or 8.1(d.3)); or

— shall be the null format, format J.

c) In all other cases, the negotiated parameter field shall be present, and shall have two sub-parts each with fixed structure and length [formats A – G, concatenated with formats J – M] as determined during the DLC establishment process (see 8.1(c.3) or 8.1(d.3)).

d) All DLEs shall support formats A, C, F, G and J.

NOTE — Support of formats A and G is required to facilitate migration of existing national standards.

e) DLEs which support both **timeliness** and a time-synchronism class (see 11.3(a)) other than **none** shall support formats K, L and M.

The various SD-parameter formats, and their potential applicability to the CA, CD, DT and ED DLPDUs, are shown in Table 48.

8.4.2.1 Parameters conveying DLCEP state and DLSDU timeliness

The formats for the first sub-part of the SD-parameters have been given names that reflect their expected usage. These formats are:

A) The **null** format, format A, is zero octets in length.

B) The **short** format, format B (Table 49), of length one octet, is as follows:

1) a one-bit retransmission-request (selective-reject) subfield, requesting (when J=1) retransmission of the NDR'th DLSDU;

2) a one-bit acknowledgment subfield, acknowledging (when K=1) that all DLSDUs before the NDR'th have been received and reported to the DLS-user;

3) a one-bit timeliness subfield, indicating (when T=1) that the associated DLSDU originated in a buffer with associated timeliness criteria, and that those timeliness criteria were met;

4) a two-bit subfield, specifying the low-order two bits of the sequence number NDR of the DLSDU being requested, or of the DLSDU after the one being acknowledged, or both;

5) a three-bit subfield, specifying the low-order three bits of the sequence number NDS of the associated DLSDU (if this DLPDU contains user data) or of the highest-numbered DLSDU which has been sent (if this DLPDU does not contain user data); and

6) the implied values for the fields of 8.4.2(c) which are omitted from 8.4.2(b) are zero.

NOTE — DLSDUs sent from a DLCEP are assigned consecutive sequence numbers, starting with one, before transmission.

C) The **long** format, format C (Tables 50, 51, and 52), of length three octets, is as follows:

1) a four-bit subfield, specifying the low-order four bits of the sequence number NDR of the DLSDU being requested, or of the DLSDU after the one being acknowledged, or both;

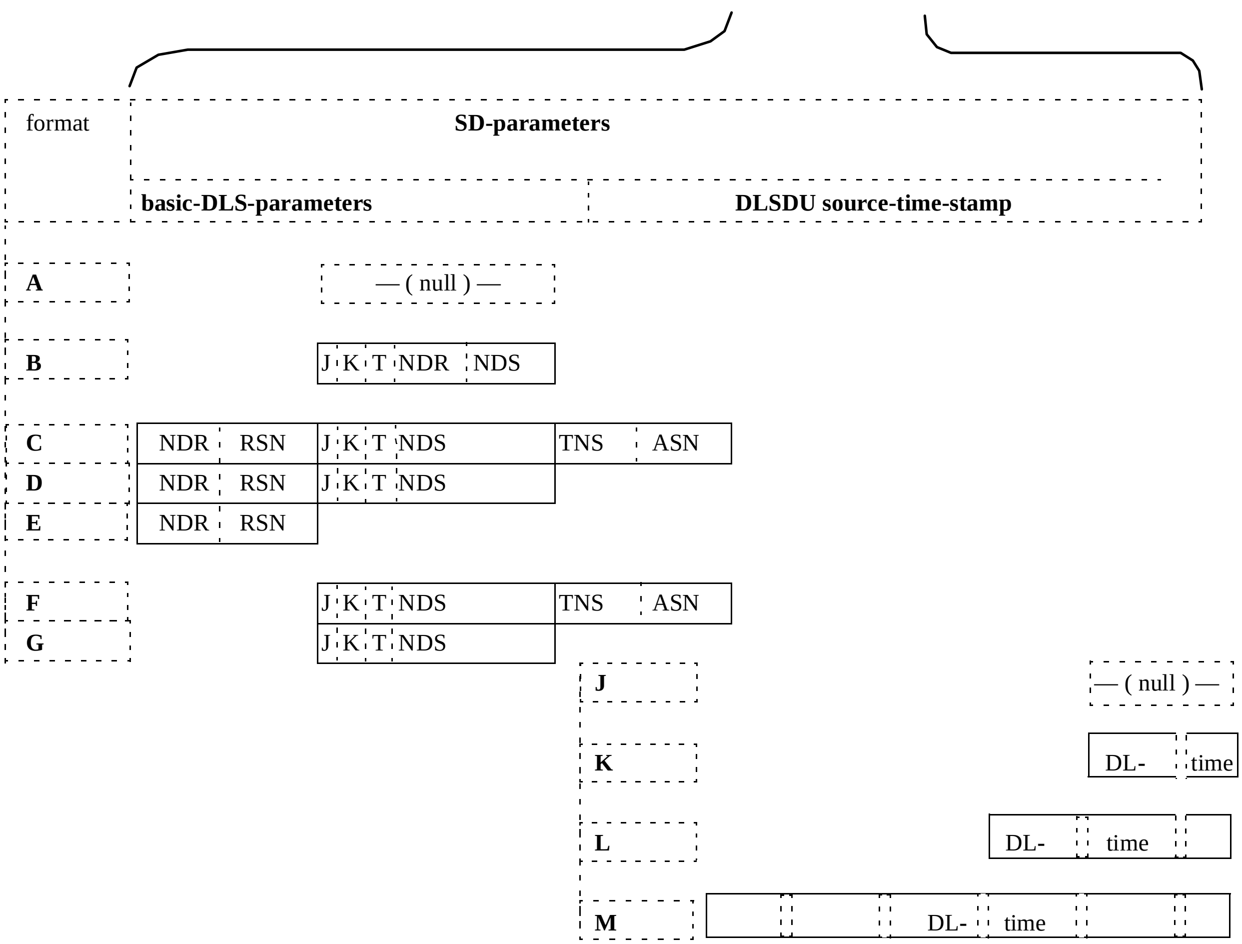
2) a four-bit subfield, specifying the zero-based segment number of the DLSDU segment being requested, or zero when no segment is being requested;

3) a one-bit retransmission-request (selective-reject) subfield, requesting (when J=1) retransmission of the NDR'th DLSDU;

4) a one-bit acknowledgment subfield, acknowledging (when K=1) that all DLSDUs before the NDR'th have been received in their entirety and reported to the DLS-user;

Table 48 — Structure of connection-oriented CA, CD, DT and ED DLPDUs

Connection-oriented DLPDUs					
format	**frame control**	**d-DLCEP-address**	**s-DLCEP-address**	**Parameters**	**user data**
1	CA 1	[HL.]N.S	[HL.]N.S	parameters	o-pDLSDU
2	CA 2	[HL.]N.S	—	parameters	o-pDLSDU
3	CD 1	[HL.]N.S	[HL.]N.S	o-parameters	—
4	CD 2	[HL.]N.S	—	o-parameters	—
5	ED 1	[HL.]N.S	[HL.]N.S	parameters	pDLSDU
6	ED 2	[HL.]N.S	—	parameters	pDLSDU
7	DT 1	[HL.]N.S	[HL.]N.S	parameters	o-pDLSDU
8	DT 2	[HL.]N.S	—	parameters	o-pDLSDU
9	DT 3		[HL.]N.S	parameters	o-pDLSDU
10	DT 4	implied PSA	—	parameters	o-pDLSDU
11	DT 5		implied PDA	parameters	o-pDLSDU



5) a one-bit timeliness subfield, indicating (when T = 1) that the associated DLSDU originated in a buffer with associated timeliness criteria, and that those timeliness criteria were met;

Table 49 — Short format SD-parameters for DLCEP state

Retransmit	Acknowledge	Timeliness	Modulus Number of Post-acknowledged or Requested DLSDU		Modulus Number of Associated DLSDU or of Highest-numbered DLSDU sent		
J	K	T	NDR mod 2^2		NDS mod 2^3		
7	6	5	4	3	2	1	0

Table 50 — Long format SD-parameters for DLCEP state: 1st octet

Modulus Number of Post-acknowledged or Requested DLSDU				Zero-based Requested Segment Number			
NDR mod 2^4				RSN			
7	6	5	4	3	2	1	0

Table 51 — Long format SD-parameters for DLCEP state: 2nd octet

Retransmit	Acknowledge	Timeliness	Modulus Number of Associated DLSDU or Highest-numbered DLSDU Sent				
J	K	T	NDS mod 2^5				
7	6	5	4	3	2	1	0

Table 52 — Long format SD-parameters for DLCEP state: 3rd octet

Zero-based Total Number of Segments in DLSDU				Zero-based Associated Segment Number			
TNS				ASN			
7	6	5	4	3	2	1	0

6) a five-bit subfield, specifying the low-order five bits of the sequence number NDS of the associated DLSDU (if this DLPDU contains user data) or of the highest-numbered DLSDU which has been sent (if this DLPDU does not contain user data);

NOTE — If this DLPDU contains no user data, then all previously-requested DLSDUs must have been sent in their entirety.

7) a four-bit subfield, specifying the zero-based total number of segments in the associated DLSDU, or zero when the DLPDU contains no user data;

8) a four-bit subfield, specifying the zero-based segment number of the associated DLSDU segment, or zero when the DLPDU contains no user data.

NOTE — This representation has the following benefits:

a) The zero-based segment number, coupled with the fact that all segments except the last are the maximum size permitted by the DLPDU's priority, facilitates DLSDU reassembly during reception.

b) The total number of segments in the actual DLSDU can be used to allocate a buffer or queue record of appropriate size upon first receipt of any segment of the DLSDU.

D) The **unsegmented long** format, format D, of length two octets, consists of the first two octets of format C (Tables 50 and 51).

E) The **subscriber** format, format E, of length one octet, consists of the first octet of format C (Table 50), with implied values for the fields J, K and T of the absent second octet. The implied values for the J, K and T fields of 8.4.2(C), which are omitted from 8.4.2(E), are one (1), zero (0) and zero (0), respectively.

F) The **publisher** format, format F, of length two octets, consists of the last two octets of format C (Tables 51 and 52). The values for the J and K fields of 8.4.2(C) shall be zero (0).

G) The **unsegmented publisher** format, format G, of length one octet, consists of the second octet of format C (Table 51). The values for the J and K fields of 8.4.2(C) shall be zero (0).

Only formats A, C, F and G need be supported; support for formats B, D and E is optional.

8.4.2.2 Parameters conveying DLSDU time-of-production

The formats for the second sub-part of the SD-parameters are:

J) The **no-time** format, format J, is zero octets in length.

K) The **2-second** format, format K, of length two octets, consists of the third-lowest-order and second-lowest-order octets of the originating DLE's DL-time at the moment of DL-Put request execution, in that order. It is adequate to distinguish the timing of events less than 2 s apart, and provides a potential resolution of 2^{-5} ms.

L) The **8-minute** format, format L, of length three octets, consists of the fourth-lowest-order, third-lowest-order and second-lowest-order octets of the originating DLE's DL-time at the moment of DL-Put request execution, in that order. It is adequate to distinguish the timing of events less than 8 min apart, and provides a potential resolution of 2^{-5} ms.

M) The **full-time** format, format M, of length six octets, consists of the high-order six octets of the originating DLE's DL-time at the moment of DL-Put request execution, encoded most-significant-octet first. It is adequate to distinguish the timing of any two events, and provides a potential resolution of 2^{-5} ms.

8.5 Structure and encoding of SR-parameters

There are two SR-parameter formats:

a) The **null** format, format X, of length zero octets, is identical to SD-parameter format P and is used to indicate that the sending bridge was able to buffer for forwarding a received CA, CD or ED

DLPDU, when the reply token was created by that received DLPDU. By using this format, the bridge implicitly returns the symbolic status “BOK” as specified in Table 53.

Table 53 — Reply status for SR DLPDUs

Short Name	**Definition**	**hexadecimal coding**
RR	failure — resource limitation in responder	4
RI	failure — responder DLE-class incompatible with this request	9
RTR	failure — LAS transfer rejected	B
BF	failure — fault in intermediary bridge	C
BR	temporary failure — resource limitation in intermediary bridge	D
BOK	interim success — intermediary bridge is forwarding transaction	E

NOTE — This table is necessarily a subset of Table 47.

b) The **error** format, format Y (Table 54), is based on SD-parameter format U (Table 46), and so is of length one octet. Format Y is used

— to indicate that a bridge which should have forwarded a received CA, CD or ED DLPDU was unable to buffer the received DLPDU, when the reply token was created by that received DLPDU (reasons BF or BR);

— to reject a CA, CD or ED DLPDU received at a DLCEP under certain error conditions; and

— to reject an attempt to transfer the LAS role to the replying DLE, when the reply token was created with a TL DLPDU (reasons RR, RI and RNC).

The subfields of the SR-parameters field’s error format are:

1) a four-bit status subfield, XXXX, encoded as specified in Table 53;

NOTE — The only two values possible for the returned status are “BF” and “BR.”

2) a four-bit subfield, ZZZZ, coded as zero and reserved for future standard use.

Table 54 — Short format SR-parameters

status				**must be zero**			
XXXX				ZZZZ			
7	6	5	4	3	2	1	0

8.6 Structure and encoding of TD-parameters

The TD-parameters field shall be composed of four subfields (Table 55):

Table 55 — Structure of TD-parameters

TD-parameters and the variable values transmitted					
field type	link originating DL-time	DL-time quality	DL-time-offset	DL-time prior to end-of-transmission	DL-time adjustment
value as sent	$V_S(TSL)$	$V_S(TQ)$	$V_S(DLTO)$	$V_S(DLTO)$ + $V_S(LSTO)$ + $C_S(NT)_1$	$P_S(TD)$ = $C_S(NT)_2$~$C_S(NT)_1$
name at receipt	$N_S(TSL)$	$N_S(TQ)$	$N_S(DLTO)$	$N_S(DLT)$	$N_S(DLTA)$
field size	two octets	one octet	seven octets	seven octets	three octets

a) The link designator of the DL-time-source DLE — the DLE which originates the sense of DL-time on the extended link — shall be expressed in two octets as specified in Annex A.

NOTE — When a bridge interconnects two or more single links into an extended link, this is the link-id of the root link of the spanning tree formed by the bridges which interconnect the links of the extended link. It is the local link-id only for DLEs on that root link, or when no bridges interconnect the local link with other links; in this latter case the link-id may have the value zero.

b) The quality of DL-time shall be expressed in one octet as follows (Table 56):

Table 56 — Structure and encoding of the DL-time-quality measures

limiting time-synchronism class			number of intervening links			time-source type	
TTT			LLL			SS	
7	6	5	4	3	2	1	0

1) a three-bit subfield, TTT, specifying the least-capable time-synchronism class (see 11.3(a)) of all of the DLEs on the DL-time propagation path from the DL-time source to the DLE, inclusive, encoded as:

000 NONE;

001 1 s;

010 100 ms;

011 10 ms;

100 1 ms;

101 100 µs;

110 10 µs;

111 1 µs;

2) a three-bit subfield, LLL, specifying the number of intervening links on the DL-time propagation path from the DL-time source DLE to the sending DLE, expressed as 0 to 7, where

0 indicates that the DL-time originates with the DLE itself;

1 indicates that the DL-time originates with another DLE on the local link;

2 indicates that the DL-time originates with a DLE on a link one bridge removed,

and so forth;

3) a two-bit subfield, SS, expressing the method of synchronization of the DL-time source DLE with Universal Coordinated Time (UTC), the world-wide time standard, encoded as:

00 some DLE's node-time, locally generated and not received from an extra-DLS-provider source;

01 local time (not UTC) received (directly or indirectly) from a human source;

10 UTC received (directly or indirectly) from a human source or an unreliable electronic source, or previously (but no longer) received (directly or indirectly) from a reliable electronic source;

11 UTC continuously received (directly or indirectly) from a reliable electronic source.

NOTES

1. Reliable electronic sources include radio receivers for national time broadcasts, atomic clocks, and similar sources which provide intrinsic worldwide time coordination.

2. The mechanisms for synchronizing a DLE which originates the extended link's sense of DL-time with an external time source is necessarily implementation-specific and beyond the scope of standardization.

c) The DL-time-offset, V_S(DLTO), of the sending DLE during DLPDU formation shall be expressed as a signed seven-octet integer encoded most-significant-octet first, where the least significant bit represents a time granularity of approximately 2^{-13} ms as shown in Table 57.

d) The DL-time, $C_S(NT)_1 + V_S(LSTO) + V_S(DLTO)$, of the sending DLE at a moment during DLPDU formation, less than 1 s prior to transmission, shall be expressed as a seven-octet non-negative integer encoded most-significant-octet first (Table 57), where the most significant bit is always zero and the least significant bit represents a time granularity of approximately 2^{-13} ms.

NOTE — The granularity of encoded DL-time does not imply that a DLE increments DL-time at that granularity, but rather that each time the DLE increments its sense of DL-time, it does so by an amount that causes the average rate of incrementation to be approximately 2^{13} counts per millisecond (which is $2^{13} \times 10^3$ counts per second).

Table 57 — Approximate numeric significance of the bits of seven-octet DL-time

octet of DL-time in transmission order	**symbolic contents**	**approximate weight of the low-order bit of the octet**
1	0YYYYYYY	1,09yr
2	DDDDDDDD	1,55day
3	HHHHHMMM	8,74min
4	MMMSSSSS	2,05s
5	Smmmmmmm	8,00ms
6	mmm•µµµµµ	31,25µs
7	µµµµµnnn	122ns

e) The time-adjustment, $P_S(TD) = C_S(NT)_2 \sim C_S(NT)_1$, is a small adjustment to account for systemic offsets in the accompanying DL-time subfield d), caused by known implementation considerations within the sending real end system when sending a TD DLPDU, such as early sampling of $C_S(NT)_1$ during TD DLPDU formation and transmission, such that the sum of the values specified by (d) and (e) will be the DL-time, $V_S(DLTO) + V_S(LSTO) + C_S(NT)_2$,

at which the END-OF-DATA-AND-ACTIVITY PhIDU (see 6.1) will be sent to the associated PhE.

The time-adjustment shall be expressed as a three-octet unsigned integer encoded most-significant-octet first (Table 58), where the least significant bit represents a time granularity of approximately 2^{-13} ms.

NOTES —

1. This field is included to avoid the requirement that the DLE perform an equivalent real-time adjustment to the full seven octet DL-time specified by (c).

2. This adjustment will vary inversely with the DLE's actual instantaneous rate of data transmission.

Table 58 — Approximate numeric significance of the bits of three-octet short time

octet of short time in transmission order	symbolic contents	approximate weight of the low-order bit of the octet
1	Smmmmmmm	8,00ms
2	mmm•µµµµµ	31,25µs
3	µµµµµnnn	122ns

NOTE — The objective of the computation of (d) and (e) is to ensure that sum of subfields d) and (e) is larger than the sender's $V_S(NTO)$ such that

A) were the sender to measure its round-trip-delay, $V_S(MD)$ (see 9.4.1.5), when communicating with itself through the PhE and medium, then

B) if the sender were to receive its own transmissions of TD DLPDUs by receiving while transmitting, then

C) the sum of subfields (d) and (e) and , as computed by the sender from the received TD DLPDU, would be equal to the current value of

$$V_S(DLTO) + V_S(LSTO) + C_S(NT)$$

at the sender at the moment that the sender's PhE reports completion of reception of the TD DLPDU from the local medium.

8.7 Structure and encoding of RQ-parameters

The RQ-parameters field shall be composed of one subfield (Table 59):

Table 59 — Structure of RQ-parameters

RQ-parameters
C_S(NT)
short-time$_1$
three octets

a) The parameter subfield shall equal the low-order 24 bits of the node-time, C_S(NT), of the sending DLE at the moment of RQ DLPDU formation just prior to (or during) transmission, representing the low-order three octets of the DLS-provider's current node-time in units of 2^{-13} ms.

NOTE — The contents of this subfield reflect the relative time at which the RQ DLPDU is transmitted.

If the sending DLE's time-synchronization class (see 11.3(a)) is NONE and the DLE does not maintain even an estimated C(NT), then it may use any value for this subfield.

Table 58 shows the transmission order and approximate numeric significance of the octets of this subfield. It is encoded as a three-octet unsigned integer sent most-significant octet first.

8.8 Structure and encoding of RR-parameters

The RR-parameters field shall be composed of four subfields (Table 60):

Table 60 — Structure of RR-parameters

RR-parameters			
$N_S(TQ)_{7-5}$	N_1(NT)	N_2(NT)	C_S(NT)
measurement quality	short-time$_1$	short-time$_2$	short-time$_3$
one octet	three octets	three octets	three octets

a) The first parameter subfield shall be encoded as specified in Table 61, and shall specify

1) the time-synchronism class of the replying DLE, TTT, encoded as in 8.6(b.1);

2) whether a DLE with a time-synchronism class of NONE maintains an estimated C(NT), E, encoded

i) as a Boolean (0=FALSE, 1=TRUE) when TTT has the value 000; and

ii) as 1 (TRUE) when TTT has a value other than 000; and

3) a 4-bit subfield, encoded as zero.

Table 61 — Structure and encoding of the RR-time-quality measures

<table>
<tr><td colspan="3">limiting time-synchronism class</td><td>DLE maintains estimated C(NT)</td><td colspan="4">reserved</td></tr>
<tr><td colspan="3">TTT</td><td>E</td><td colspan="4">0000</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

b) The second and third parameter subfields shall equal the received and locally-appended parameter subfields, respectively, of the received RQ DLPDU.

c) The fourth parameter subfield shall equal the low-order 24 bits of the node-time, $C_S(NT)$, of the replying DLE at the moment of RR DLPDU formation just prior to (or during) transmission, representing the low-order three octets of the DLS-provider's current node-time in units of 2^{-13} ms.

NOTE — The contents of this subfield reflect the relative time at which the RR DLPDU is transmitted. The resulting round-trip-delay computations are accurate, independent of any internal delays within the DLEs, provided that the sum of the propagation delays between the two DLEs is less than 2 s.

If the sending DLE's time-synchronization class (see 11.3(a)) is NONE and the DLE does not maintain even an estimated C(NT), then it shall encode this subfield as zero.

Table 58 shows the transmission order and approximate numeric significance of the octets of each of the last three subfields. Each of these subfields is encoded as a three-octet unsigned integer sent most-significant octet first.

8.9 Structure and encoding of PN-parameters

The PN-parameters field specifies the current values of those DLE and PhE parameters necessary for a receiving DLE to configure itself and its associated PhE so that they can reply to a subsequent PN DLPDU. Once set, the values of these parameters can not be changed while the receiving DLE remains ONLINE, other than by reception of another PN DLPDU as specified in 10.1.3.

NOTE — Theoretically, the current LAS DLE could change the link's parameters by sending a PN DLPDU to each of the other DLEs on the local link, forcing each of them OFFLINE, after which they would learn the new link parameters in the process of again coming ONLINE. But this DL-protocol does not specify a means of requesting this LAS action.

The PN-parameters field shall be composed of seven subfields (Tables 62 through 67):

Table 62 — Structure of PN-parameters

PN-parameters and the variable values transmitted						
PhL maximum inter-channel signal skew	version	PhL post-trans-mission-gap extension units	PhL preamble extension units	slot-time	maximum-response-delay	minimum-inter-DLPDU-delay
V(PhIS)	0VVV	V(PhGE)	V(PhPE)	V(ST)	V(MRD)	V(MID)
one quartet	one quartet	one quartet	one quartet	two octets	one octet	one octet

The PN-parameters field, which partitions as shown in Table 62, shall specify the parameters needed to reply to the PN DLPDU:

a) The first two octets, ordered as shown in Tables 63 and 64, specify the DL protocol version number and the PhL-parameters of the local link, defined in Table 1 of Part 2 of this International Standard, required for generating a reply to a received PN DLPDU:

Table 63 — PN-parameters: 1st octet

PhL maximum inter-channel signal-skew				**zero**	**version**		
V(PhIS)				0	VVV		
7	6	5	4	3	2	1	0

1) a 4-bit subfield which specifies the required PhL maximum inter-channel signal skew, defined in 6.2 and 8.4.2(e) of Part 2 of this International Standard, for the PhEs of the local link;

2) a 1-bit subfield, encoded as zero;

3) a 3-bit DL protocol version number (VVV), whose value is specified in 8;

Table 64 — PN-parameters: 2nd octet

PhL post-transmission-gap extension units				**PhL preamble extension units**			
V(PhGE)				V(PhPE)			
7	6	5	4	3	2	1	0

4) a 4-bit subfield which specifies the required number of PhL post-transmission-gap extension units, defined in 6.2 and 8.4.1(d) of Part 2 of this International Standard, for the PhEs of the local link;

NOTE – Although this measure is named “post-transmission-gap -extension” in Part 2 of this International Standard, it is a measure of the amount of observable non-transmission required between any two transmissions on the local medium, as observed from any point on that medium, whether the two transmissions are from the same source PhE or from two different source PhEs. Thus it could just as correctly be named “pre-transmission-gap-extension,” or even more correctly “inter-transmission-gap-extension.”

5) a 4-bit subfield which specifies the required number of PhL preamble extension units, defined in 6.2 and 8.4.1(c) of Part 2 of this International Standard, for the PhEs of the local link;

b) The next four octets, ordered as shown in Tables 65 through 67, specify the DLL-parameters for the local link required for generating a reply to a received PN DLPDU:

Table 65 — PN-parameters: 3rd and 4th octets

slot-time															
V(ST)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 66 — PN-parameters: 5th octet

maximum-response-delay							
V(MRD)							
7	6	5	4	3	2	1	0

Table 67 — PN-parameters: 6th octet

minimum-inter-DLPDU-delay							
V(MID)							
7	6	5	4	3	2	1	0

1) The fifth parameter subfield shall convey, in two octets transmitted most significant octet first, the current value of V(ST), slot-time, defined in 5.7.1.1.

2) The sixth parameter subfield shall convey, in one octet, the current value of V(MRD), maximum-response-delay, defined in 5.7.1.3.

3) The seventh parameter subfield shall convey, in one octet, the current value of V(MID), minimum-inter-DLPDU-delay, defined in 5.7.1.12.

8.10 Structure and encoding of DD-parameters

The DD-parameters (Delegation-Duration parameters) field shall be composed of one subfield (Table 68):

Table 68 — Structure of DD-parameters

DD-parameters
delegated or requested duration
V(RD)
two octets

a) The subfield shall specify the duration for which the token is being delegated or for which delegation is being requested. It is measured in octet-durations, and is encoded as a two-octet unsigned integer sent most-significant octet first. Its range is 0 to 65 000.

NOTE — The minimum practical value for using this duration for a transmission is about 16. The value 0 is used only during maintenance of the local link's live-list V(LL).

9 DL-service elements of procedure

Throughout this clause, the value (V(NRC)+1) is used as a link-independent indication of the maximum number of times to perform a procedure — once initially, plus V(NRC) repetitions — before concluding that an unrecoverable error situation exists and that alternate more-drastic action is required. When the DLE can ascertain, from the involved DL(SAP)-addresses or DLCEP-addresses, that all of the DLEs involved in a DLC are local to a single link, then the DLE may substitute in a consistent manner the link-dependent value (V(MRC)+1) for the link-independent value (V(NRC)+1).

NOTE — Throughout this clause, a DLE receives a transmit opportunity, either by receipt of an appropriate immediate-reply opportunity or by receipt of an appropriate token.

9.1 Operation of the DL(SAP)-address, buffer and queue management services

The DL(SAP)-address, buffer and queue management services are the create and delete buffer or queue services, the bind and unbind DL(SAP)-address services, the put buffer service, and the get buffer or queue service.

9.1.1 Receipt of a DL-CREATE request primitive

When the DLE receives a DL-CREATE request, it shall

a) allocate a buffer of the specified DLSDU size, or queue of the specified DLSDU size and maximum number of entries, and initialize it to empty;

NOTE — Statistical allocation techniques using a shared multi-use storage area are permissible.

b) assign a buffer-or-queue DL-identifier, and any provided DLS-user-identifier, to that buffer or queue; and

c) return that identifier to the DLS-user, together with a status of “success.”

Alternatively, an appropriate failure status shall be returned to the DLS-user.

9.1.2 Receipt of a DL-DELETE request primitive

When the DLE receives a DL-DELETE request, if the specified buffer or queue

a) was created by the DLS-user and not by DL-management action; and

b) is not currently bound to any DLCEP or DL(SAP)-address

then the DLE shall delete the specified buffer or queue and return a status of “success.”

Otherwise the DLE shall return an appropriate failure status.

9.1.3 Receipt of a DL-BIND request primitive

When the DLE receives a DL-BIND request, it shall check

a) that the specified DL(SAP)-address is a group DL-address not currently in use by the requesting DLS-user, or a DLSAP-address not currently in use by the DLE;

b) that the specified link and node components of the DL(SAP)-address are compatible with the one or more link address components and one or more node address components assigned to the DLE (see Annex A);

c) that any specified explicit bindings to DL-queues or DL-buffers are valid and do not conflict with other existing bindings to those queues or buffers; and

d) that any specified static or dynamic QoS attributes are valid and permitted by local DL-management.

Failure shall be reported to the DLS-user with an appropriate status. Otherwise

1) the DL(SAP)-address shall be associated to the requesting DLS-user, and to any provided DLS-user-identifier;

2) all explicitly-specified queues and buffers shall be bound as specified to the DL(SAP)-address;

3) the static and dynamic QoS attributes shall be bound to the DLSAP-address;

4) the specified DL(SAP)-address shall be activated for reception;

5) a DL(SAP)-address identifier shall be assigned to the DL(SAP)-address and returned to the DLS-user, together with a status of "success;" and

6) if either

i) the DLE is a fractional-duty-cycle (FDC) DLE; or

ii) the DL(SAP)-address has a link-designator component (see Annex A/A.2.1) whose value specifies a flat non-local DL-address (between 0001_{16} and $0FFF_{16}$, inclusive), and the variable V(TL) has a non-zero value

then the DLE shall send a DL-address report SPDU as specified in Annex B/B.3.6.3, with an appropriate reason for the report SPDU and with a request identifier of zero, to the DL-support functions of all bridges on the local link (see Annex A/A.3.2).

9.1.4 Receipt of a DL-UNBIND request primitive

When the DLE receives a DL-UNBIND request,

a) if the specified DL(SAP)-address currently is not bound to the requesting DLS-user, or if the specified DL(SAP)-address was bound by DL-management action, then the DLE shall ignore the request primitive;

b) otherwise the DLE shall

— disconnect any DLCEPs associated with the DL(SAP)-address;

— confirm with an appropriate error status any unconfirmed connection-oriented or connectionless-mode service requests outstanding at that DL(SAP)-address or its DLCEPs;

— unbind all queues and buffers from the specified DL-address; and

— disassociate the specified DL-address from the requesting DLS-user.

If the DLE has no other DLS-users associated with that DL-address (which is always the case for DLSAP-addresses) then the DLE shall deactivate reception of that DL-address, in which case if either

1) the DLE is a fractional-duty-cycle (FDC) DLE; or

2) the DL(SAP)-address has a link-designator component (Annex A/A.2.1) whose value specifies a flat non-local DL-address (between 0001_{16} and $0FFF_{16}$, inclusive), and the variable V(TL) has a non-zero value

then the DLE shall send a DL-address report SPDU as specified in Annex B/B.3.6.3, with an appropriate reason for the report SPDU and with a request identifier of zero, to the DL-support functions of all bridges on the local link (see Annex A/A.3.2).

9.1.5 Receipt of a DL-PUT request primitive

When the DLE receives a DL-PUT request for a buffer

a) which is associated with the requesting DLS-user; and

b) which is not bound as a receiving buffer

1) to a peer or subscriber DLCEP; or

2) to a DLSAP-address whose DL(SAP)-role is INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER; and

then

c) if a DLSDU is presented, and the size of the DLSDU is less than or equal to the buffer-size, then the DLE shall

1) set the contents of the buffer equal to the presented DLSDU;

2) indicate for each DLCEP which has a sending binding to the buffer that the buffer contains a new DLSDU;

3) set the timeliness-status of buffer-writing, $V_B(TS)$ (see 5.7.4.21), to the DLS-user-specified timeliness, or FALSE if the user did not specify timeliness;

4) if the timeliness-status of buffer-writing is TRUE, then set the variable time-of-last-buffer-write, $V_B(TW)$ (see 5.7.4.19), associated with the buffer to the current DL-time;

5) if the timeliness-status of buffer-writing is TRUE, then set the time-of-production associated with the buffer, $V_B(TP)$ (see 5.7.4.20), to the current DL-time; and

6) return a status of “success;”

d) else if (c) does not apply, and no DLSDU is being presented (that is, the buffer is being set empty), and the buffer is bound to a DLSAP-address, then the DLE shall set the buffer empty and shall return a status of “success;”

e) else if neither (c) nor (d) applies, then the DLE shall return an appropriate failure status.

Any ongoing accesses to the contents of a buffer or to its associated timeliness information, which are incomplete at the time of a DL-PUT request, shall not be affected by the DL-PUT request.

NOTE — This constraint ensures that each access to a buffer is logically atomic.

9.1.6 Receipt of a DL-GET request primitive

a) If the DLE receives a DL-GET request for a buffer which is associated with the requesting DLS-user, then

1) If the buffer was written by a DLCEP which specified RESIDENCE timeliness or UPDATE timeliness or SYNCHRONIZED timeliness or TRANSPARENT timeliness, then the DLE shall evaluate the associated timeliness criteria as specified in 9.1.7, using the current DL-time as the time of buffer readout, and shall return the result as the local-DLE-timeliness attribute of the request.

2) Otherwise, if (1) does not apply, then the DLE shall return the value FALSE as the local-DLE-timeliness attribute of the request.

3) The DLE shall return the timeliness-status, $V_B(TS)$ (see 5.7.4.21), associated with the buffer's writing as the sender-and-remote-DLE-timeliness attribute (see 9.2.2.5.3) of the request.

NOTE — This sender-and-remote timeliness attribute always has the value FALSE when the buffer was written by an instance of the unitdata exchange service.

4) If the sender-and-remote-DLE-timeliness attribute associated with the buffer is TRUE, and the buffer was written by a DLCEP which provides DL-time-of-production, then the DLE shall return the DL-time-of-production associated with the buffer, $V_B(TP)$ (see 5.7.4.20).

5) The DLE shall return the current contents of the buffer, with a status of "success" if the buffer is non-empty, and a status of "possible failure — buffer empty" if the buffer is empty.

6) If the buffer is a non-retentive buffer (BUFFER-NR), then the buffer shall be set empty.

7) Reading a buffer and its associated timeliness information shall be logically atomic with respect to writing the buffer.

Implementor's NOTE — When an implementation provides access to a buffer for an extended period of time during buffer read or write, then in the worst case this atomicity restriction requires that separate copies of the buffer's contents and timeliness information be provided for each reader and for the one writer, in addition to the actual buffer with its contents and timeliness. Then each reader may be in the middle of an extended access to a different epoch of the buffer's contents, and the writer may be writing a tentative buffer during reception, which will become the current buffer only if a Ph-error or FCS error is not detected before the end of the reception process.

b) If the DLE receives a DL-GET request for a DLS-user specified queue which is

— associated with the requesting DLS-user;

— not bound as a sending queue either to a DLCEP or to a DLSAP-address; and

— non-empty

then the DLE shall

1) return the next DLSDU contained in the queue, together with the called DLC identifier associated with that DLSDU, or DLL priority and called and calling DL(SAP)-addresses or DL(SAP)-address-identifiers associated with that DLSDU;

2) return local- and remote-DLE-timeliness attributes of FALSE;

3) remove that DLSDU from the queue; and

4) return a status of "success."

Otherwise the DLE shall return an appropriate failure status.

9.1.7 Computation of DL-timeliness

The DLE shall compute the intrinsic DL-timeliness for the buffer based on the type of buffer access — writing or reading — and the corresponding type of DL-timeliness which was specified on the corresponding local DL-CONNECT request or response primitive, as follows, where $P_C(NP.\Delta T)$ is the appropriate time-window-size specified in that request or response primitive:

a) **Residence**

DL-timeliness $\equiv$ TRUE when $0 \leq$ (current-DL-time – $V_B(TW)$)
and (current-DL-time – $V_B(TW)$) $\leq P_C(NP.\Delta T)$;

DL-timeliness $\equiv$ FALSE otherwise.

b) **Update**

DL-timeliness $\equiv$ TRUE when $0 \leq (V_B(TW) - V_C(TNA))$
and $(V_B(TW) - V_C(TNA)) \leq P_C(NP.\Delta T)$;

DL-timeliness $\equiv$ FALSE otherwise.

c) **Synchronized**

DL-timeliness $\equiv$ TRUE when $0 \leq (V_B(TW) - V_C(TNA))$
and $(V_B(TW) - V_C(TNA)) \leq$ (current-DL-time – $V_C(TNA)$)
and (current-DL-time – $V_C(TNA)$) $\leq P_C(NP.\Delta T)$;

DL-timeliness $\equiv$ FALSE otherwise.

d) **Transparent**

DL-timeliness $\equiv$ TRUE.

e) **None**

DL-timeliness $\equiv$ FALSE.

9.2 Operation of the connection-mode services

The connection-mode services are the DLCEP establishment and DLCEP release services, the DLC data transfer and DLCEP reset service, and the DLC subscriber query service.

Figure 16 of Part 3 of this international standard shows a conceptual state transition diagram for sequences of DLC service primitives at a DLCEP. A corresponding state transition diagram for this DL-protocol is shown in Figure 7 and in Annex F/F.2.1, where similar states have identical state numbers.

The primitives of the DLCEP-establishment service are DL-Connect request, indication, response and confirm, and DL-Connection-Established indication. The primitives of the DLCEP-release service are DL-Disconnect request and indication. The associated local DLS-user, or the DLCEP's remote peer or publisher DLS-user, or the DLS provider, may invoke the DLCEP-release service any time after DLCEP establishment has begun.

During DLCEP establishment and release, if the DLCEP is a peer or publisher DLCEP, and the DLCEP-address used by the DLCEP is not sent as a source DL-address in a DC or EC DLPDU, and if either

a) the DLE is a fractional-duty-cycle (FDC) DLE; or

b) the DLCEP-address has a link-designator component (see Annex A/A.2.1) whose value specifies a flat non-local DL-address (between 0001_{16} and $0FFF_{16}$, inclusive), and the variable V(TL) has a non-zero value

then the DLE shall send a DL-address report SPDU as specified in Annex B/B.3.6.3, including the specified DLCEP-address, with an appropriate reason for the report SPDU and with a request identifier of zero, to the DL-support functions of all bridges on the local link (see Annex A/A.3.2).

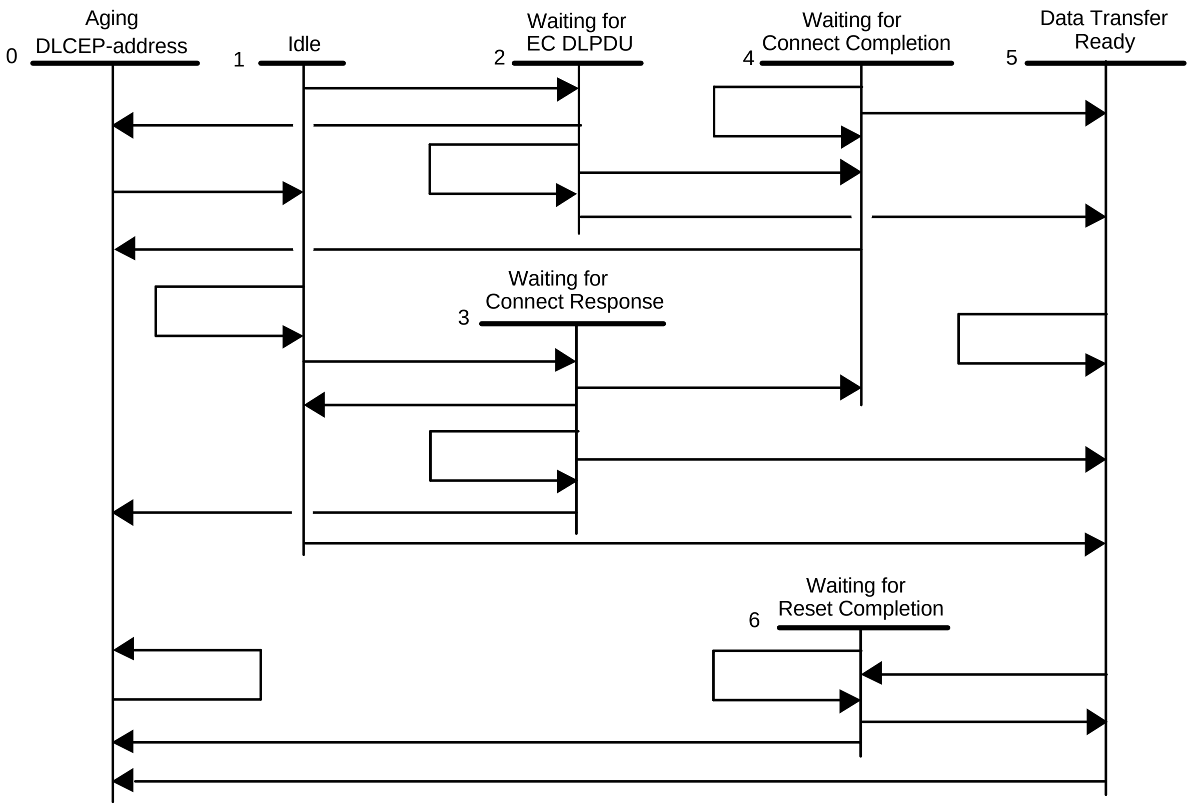


Figure 7 — State transition diagram for a DLCEP

9.2.1 Operation of the DLCEP establishment and DLCEP release services

9.2.1.1 DLC negotiation rules

The DLS-user-visible aspects of the DLC negotiation rules are specified in Part 3 of this International Standard. Additional negotiation rules which do not impact DLS-user visible aspects of the DLC are specified in 8.1 of this part of this International Standard. In case of apparent conflict, the rules specified in this sub-clause take precedence over those specified in 8.1, which in turn take precedence over those specified in Part 3 of this International Standard:

NOTE – Where a subscribing DLS-user which is attempting to join an existing DLC requests DLC attributes which are not provided by the attributes of that existing DLC, then the publishing DLS-user may either

A) indicate its inability to provide the requested attributes by replying with a DL-DISCONNECT request primitive to the proposed subscriber;

B) create a new DLC with the desired attributes and connect the subscriber to that DLC; or

C) modify the existing DLC to include the requested characteristics, when permitted by this part of this International Standard.

a) If the publisher, or either peer, of a DLC specifies a DLPDU-authentication attribute of MAXIMAL, then

NOTE — DLPDU-authentication of MAXIMAL is provided primarily for use in safety systems. For this reason it maximizes the amount of state information exchanged in each DLPDU sent on the DLC and prohibits two-way -user-data exchange in a single transaction, centralized schedule execution and other activities in which multiple DLEs need to have consistent state information.

1) each DLPDU sent from each DLCEP of the DLC shall contain the maximum permitted number of explicit addresses;

2) the EC-parameters in each EC DLPDU shall be constrained as follows:

i) the address-size subfield (SS) shall specify LONG;

ii) the DLPDU-authentication subfield (XX) shall specify MAXIMAL;

iii) the residual-activity subfield (A) shall specify TRUE in the publisher-to-subscriber direction, or in all sending peer-to-peer directions, of data transfer;

NOTE — Residual activity is not meaningful in the subscribers-to-publisher direction.

iv) both 2-way data exchange subfields (E) shall specify FALSE; and

v) SD-parameter format B (subfield FFF), and time-stamp formats K and L (subfield HH), shall not be requested or used in either direction on the DLC; and

3) if a DL-SCHEDULE-SEQUENCE request specifies a sequence in which one or more sequence elements specify the local or remote DLCEP of this DLC, then that sequence may be either locally or centrally scheduled, but shall be locally executed (see 9.4.3(a), (c), (d), or (f)).

b) If (a) does not apply then

i) if the publisher, or either peer, of a DLC specifies a DLPDU-authentication attribute of SOURCE, then the DLPDU-authentication subfield (XX) in the EC-parameters shall specify SOURCE, and each DLPDU sent from each DLCEP of the DLC shall contain the maximum permitted number of explicit addresses;

ii) if a subscriber of a DLC specifies a DLPDU-authentication attribute of MAXIMAL in a DL-CONNECT request primitive, then the DLPDU-authentication subfield (XX) in the resulting EC-parameters DLPDU shall specify MAXIMAL. If the requested DLC was already established, then

A) if that DLC was not established with DLPDU-authentication attribute of MAXIMAL then the publishing DLE shall reject the connection establishment request from that subscriber;

B) otherwise, when (A) does not apply, then the publishing DLE shall attempt to add that subscriber to the existing DLC;

iii) if a subscriber of a DLC specifies a DLPDU-authentication attribute of SOURCE in a DL-CONNECT request primitive, then the DLPDU-authentication subfield (XX) in the resulting EC-parameters DLPDU shall specify SOURCE. If the requested DLC was already established, then

A) if that DLC was established with a DLPDU-authentication attribute of ORDINARY then the publisher's DLE shall change the DLPDU-authentication to SOURCE and each DLPDU sent from each DLCEP of the DLC shall thereafter contain the maximum permitted number of explicit addresses;

B) otherwise, when (A) does not apply, then the publishing DLE shall attempt to add that subscriber to the existing DLC;

iv) else if none of (i) through (iii) apply, then the DLPDU-authentication subfield (XX) in the EC-parameters shall specify ORDINARY, and each DLPDU sent from each DLCEP of the DLC shall contain the minimum permitted number of explicit addresses;

c) The DLL path-diversity subfield (Q) of the EC-parameters shall specify ANY-PATH as a default. Negotiation of this subfield is from ANY-PATH to THIS-PATH.

NOTE — See the NOTE after 8.1(a.3) for use of the value THIS-PATH and choice of the actual path in the THIS-PATH case.

d) The address-size subfield of the EC-parameters shall be determined as follows:

1) If required by (a.2.i), or if any of the DL-addresses of the EC DLPDU have only a LONG representation, then the address-size subfield of the EC-parameters shall specify LONG.

NOTE — A DL-address has a SHORT representation when there is an equivalent two-octet DL-address (see 6.2.1.1).

2) Else, when (1) does not apply, and either (b.i) applies or any member of the DLC is a fractional-duty-cycle (FDC) DLE, then the address-size subfield of the EC-parameters shall specify SHORT.

3) Otherwise, when (1) and (2) do not apply, the address-size subfield of the EC-parameters shall specify VERY-SHORT.

NOTE — The address-size VERY-SHORT applies only to DT DLPDUs sent using a reply token (see 7.7.1 formats 4 and 5); in all other cases the address-size SHORT is actually used.

e) The DLCEP-data-delivery-features subfield (TT) of the EC-parameters shall specify, independently for each direction of the DLC, the provided data-delivery features, as specified in Part 3 of this International Standard, except that the value NONE shall be replaced by UNORDERED with a maximum window size (WWWW) of zero and maximum-DLSDU-size subfield (M…M) of zero in the corresponding direction, indicating a simplex DLC.

f) The residual-activity subfield (A) of the EC-parameters shall specify TRUE in a publisher-to-subscriber or sending peer-to-peer direction of data transfer when so required by (a.2.iii), or by DL-management, or by a publishing or peer DLS-user, and shall specify FALSE otherwise Negotiation of this subfield is from FALSE to TRUE.

g) Window size negotiation occurs independently for each direction of the DLC. The actual maximum window size for a given direction of transmission shall be the smaller of the sender's maximum window size and the receiver's maximum window size in that direction, and the maximum-window-size subfield (WWWW) of the EC-parameters shall specify zero only when the maximum-DLSDU-size subfield (M…M) in the same direction is zero, indicating a simplex DLC.

h) The SD-parameter-format subfield (FFF) of the EC-parameters shall specify the negotiated format for each direction of data transmission. The initial formats for the two directions of data transmission shall be chosen to meet the following constraints:

NOTE 1 — These constraints are imposed by the DLC protocol and selected service features.

1) If (a) applies, then format B shall not be chosen for either direction.

2) If the DLCEP-data-delivery-features subfield (TT) in either direction specifies ORDERED, DISORDERED, or CLASSICAL, then

i) if the maximum-window-size subfield (WWWW) in that direction specifies a value of four (4) or more, then format B shall not be chosen for that direction;

ii) if the DLCEP-class subfield (CC) specifies PEER, then the format chosen for the other direction shall contain both J and K subfields;

iii) if the DLCEP-class subfield (CC) specifies PUBLISHER, then the format chosen for that direction shall be format F or G, and the format chosen for the other direction shall be format C or D or E;

iv) if the DLCEP-class subfield (CC) specifies SUBSCRIBER, then the format chosen for that direction shall be format C or D or E, and the format chosen for the other direction shall be format F or G.

NOTE — Format C is included in (iii) and (iv) only because support of formats D and E are not mandatory.

3) If a DLCEP-data-delivery-features subfield (TT) specifies ORDERED, DISORDERED, or CLASSICAL, then the format chosen for that direction shall contain an NDS subfield, and the format chosen for the other direction shall contain an NDR subfield.

4) If a maximum-DLSDU-size subfield (M…M) specifies a value greater than the amount of DLS-user-data that can be conveyed in a single DLPDU of the DLC's priority, as determined by

the DLC's DLL priority subfield (PP), then the format chosen for that direction shall contain TNS and ASN subfields, and the format chosen for the other direction shall contain an RSN subfield.

5) If a timeliness-included subfield (G) specifies TRUE, then the format chosen for that direction shall contain a T subfield.

6) The format chosen shall be the shortest (fewest octets in length) possible of those which the DLE supports and which meets all of constraints (1) through (5).

NOTE 2 — The expected set of formats in the sending direction are as follows:

a) formats A, and potentially E, from a PEER DLCEP when the sending direction's DLCEP-data-delivery-features subfield (TT) specifies UNORDERED, depending on the corresponding DLCEP-data-delivery-features subfield (TT) in the other direction;

b) formats B, C and D, and potentially F and G, from a PEER DLCEP when the sending direction's DLCEP-data-delivery-features subfield (TT) specifies ORDERED, DISORDERED, or CLASSICAL, depending on the corresponding DLCEP-data-delivery-features subfield (TT) in the other direction;

c) formats A, F and G from a PUBLISHER DLCEP; and

d) formats A, C, D, E and G from a SUBSCRIBER DLCEP, where formats D and G are chosen when subscribers-to-publisher DLSDUs have associated timeliness, and format C is chosen when format D would otherwise be chosen but is unavailable because its support is not mandatory. For formats C, D and G, the publishing DLE will ignore the NDS and K subfields within the SD-parameters of received CA, CD, DT and ED DLPDUs.

i) The 2-way-data-exchange subfield (E) shall specify FALSE when the maximum-DLSDU-size subfields (M…M) is zero in one direction of data transmission, or when requested by a peer DLE during negotiation of a PEER DLC, or when required by (a.2.iv) or by local DL-management or by DLE construction; and shall specify TRUE otherwise Negotiation of this subfield is from TRUE to FALSE.

j) Timeliness attributes of the DLCEP are communicated but not negotiated:

1) The timeliness-included subfield (G) of the EC-parameters shall specify FALSE when the specified sender-timeliness is NONE, and shall specify TRUE otherwise.

2) The time-stamp-format subfield (HH) of the EC-parameters shall specify

i) format J when there is no sender timeliness, or the specified sender-timeliness is NONE, or time-of-production is not requested;

ii) format K when there is sender timeliness, and time-of-production is requested, and the timeliness-class is RESIDENCE or SYNCHRONIZED, and the associated time-window-size (ΔT) is 1 s or less;

iii) format L when there is sender timeliness, and time-of-production is requested, and the timeliness-class is RESIDENCE or SYNCHRONIZED, and the associated time-window-size (ΔT) is greater than 1 s; and

iv) format M when there is sender timeliness, and time-of-production is requested, and the timeliness-class is UPDATE or TRANSPARENT.

k) If one direction of data communication is not required for the DLC, because the DLS-user-specified data delivery features for that direction specified NONE, then in that direction

1) the residual-activity subfield (A) shall be specified as FALSE;

2) the Queue/Buffer (B) subfield shall be specified as QUEUE;

3) the timeliness subfield (G) shall be specified as FALSE; and

4) the time-stamp-format subfield (HH) shall be specified as FORMAT J.

9.2.1.2 Receipt of a DL-CONNECT request primitive

When the DLE receives a DL-CONNECT request primitive from a DLS-user, the DLE shall perform the following series of actions, and if any error is detected during the process, then the DLCEP shall be disconnected as specified in 9.2.1.8:

a) The DLE shall assign a new DLCEP-identifier, and the provided DLS-user-identifier, to the DLCEP which may result from the request, and provide that DLCEP-identifier to the DLS-user as the single output parameter of the request.

b) The DLE shall create and start a user-request timer T_U(MCD) with a duration based on the user-specified maximum confirm delay for DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY primitives. If the specified value was other than UNLIMITED, then the duration of this timer should be ; otherwise the duration should be . DL-management may override these preferred durations.

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

c) The DLE shall validate the calling-DLSAP-address or DLCEP-identifier provided by the DLS-user; if invalid, the DLE shall reject the DL-CONNECT request with a DL-DISCONNECT indication.

d) The DLE shall validate the self-consistency of the requested QoS parameter set, where all static and non-specified dynamic parameters assume the default values associated with the calling-DLSAP-address, and where the following automatic adjustments to that QoS occur:

1) Where any parameter is in violation of a local DL-management-imposed limit, then that parameter shall be set equal to that limit, if permitted by the negotiation rules specified in 9.2.1.1, or the DLE shall reject the DL-CONNECT request with a DL-DISCONNECT indication.

2) If any maximum DLSDU size equals zero or the sending DLCEP data delivery features specify NONE, then the corresponding sending DLCEP data delivery features shall be set to UNORDERED.

NOTE — This special case is not considered to be a violation of the negotiation rules of 9.2.1.1.

3) If any maximum DLSDU size is greater than the maximum which can be conveyed in a single DLPDU of the specified DLL priority (see 5.2.3), and the sending DLCEP data delivery features in the corresponding direction are UNORDERED, then those sending DLCEP data delivery features shall be set to ORDERED.

e) If the calling address identifier is a DLCEP-identifier for an existing DLCEP, then

1) If the existing DLCEP is a publisher DLCEP, then the DLE shall

i) use the specified DLCEP's local DLSAP-address as the calling-DLSAP-address for this new DL-CONNECT request;

ii) set each QoS parameter equal to the corresponding parameter of the specified DLCEP, if permitted by the negotiation rules of 9.2.1.1, or the proposed DL-CONNECT request shall be rejected with a DL-DISCONNECT indication specifying “connection rejection — QoS not available, permanent condition,” terminating processing of the DL-CONNECT request;

iii) use the specified DLCEP's publisher-DLCEP-address reuse-discriminator in the EC-parameter; and

iv) consider the new DL-CONNECT request to be a request for adding new subscribers to the existing DLC and continue with (j).

2) If the existing DLCEP's DLCEP-class is PEER or SUBSCRIBER, then the DLE shall reject the DL-CONNECT request with a DL-DISCONNECT indication.

f) Otherwise, if (e) does not apply, then the DLE shall determine the maximum send and receive window sizes based on the respective buffer-and-queue bindings, as follows

1) If the DLCEP-features are NONE, then the corresponding window size shall be zero (0).

2) Otherwise, if (1) does not apply, then

i) if the binding was to a BUFFER, then the corresponding window size shall be one (1);

ii) if the binding was to a QUEUE-*K*, then the corresponding window size shall be the smaller of *K* or 15;

iii) if the default binding was used, then the corresponding window size shall be at least one (1); and

iv) in all cases, DL-management can further constrain this window size.

g) If the optional calling-DLCEP-address was specified in the request primitive, then the DLE shall assign that DLCEP-address to the DLCEP; if any conflicting assignment is detected, then the DLCEP shall be terminated as specified in 9.2.1.8.

h) Otherwise, if (g) does not apply, and if the called address is not a DLCEP-address presumed to be for a publisher DLCEP, then a DLCEP-address allocated to the DLE and not currently in use shall be assigned to the DLCEP; where possible this shall be a SHORT DL-address of the local link.

When the called address is a DLCEP-address presumed to be for a publisher DLCEP, then this assignment is optional.

i) The DLC shall initialize the DLCEP's $V_C(NP)$, $V_C(N)$, $V_C(R)$, $V_C(A)$, $V_C(M)$, $V_C(MS)$, $V_C(H)$, $V_C(HS)$ and $V_C(L)$ variables as specified in 5.7.4.

j) The DLE shall encode an EC DLPDU as specified in 7.1 and 8.1:

1) If the called-DL-address parameter specifies a DL(SAP)-address or DLCEP-address, then the DLE shall form an EC DLPDU with three addresses, whose values shall be, respectively,

i) the called-DL(SAP)-address or DLCEP-address;

ii) the DLCEP-address assigned to the DLCEP, or calling-DLSAP-address if no such assignment was done as in (h);

iii) the calling-DLSAP-address.

2) If the called-DL(SAP)-address parameter specifies UNKNOWN, then

i) if the DLS-user did not specify a calling-DLCEP-address, then the DLE shall reject the DL-CONNECT request with a DL-DISCONNECT indication;

ii) else if (i) does not apply, and the DLCEP-class is PUBLISHER, then the DLE shall form an EC DLPDU with two addresses, whose values shall be the DLCEP-address assigned to the DLCEP, and the calling-DLSAP-address, respectively;

iii) else if (i) does not apply, and the DLCEP-class is PEER or SUBSCRIBER, then the DLE shall not form and send an EC DLPDU, but shall await the receipt of a complementary EC DLPDU from the remote PEER or PUBLISHER.

k) If the DLCEP class of the DLE is to be either PEER or SUBSCRIBER, then

1) if an EC DLPDU was formed, then

i) the DLE shall set the reply-requested field in the EC-parameters in the DLPDU; and

ii) the DLE shall queue the DLPDU at TIME-AVAILABLE priority as specified in 9.4.5.

2) the DLE shall activate recognition of the DLCEP's local DLCEP-address and change the DLCEP state, $V_C(ST)$, to WAITING-FOR-EC-DLPDU.

l) If the sending DLCEP class of the DLE is to be PUBLISHER, then

1) The DLE shall clear the reply-requested field in the EC-parameters in the DLPDU.

2) If the source DLCEP-address is not that of an existing DLCEP, then the DLE shall assign a new value to the publisher-DLCEP-address reuse-discriminator subfield (NNN) of the EC-parameters (see 8.1(a.2)):

i) If the DLE is capable of recording the publisher-DLCEP-address reuse-discriminator between DLCEP incarnations, then it should maximize the interval between reuse of the same discriminator value;

ii) Otherwise, when (i) does not apply, the DLE shall choose the discriminator value randomly.

3) The DLE shall queue the DLPDU at TIME-AVAILABLE priority as specified in 9.4.5.

4) The DLE shall issue the DL-CONNECT confirm for the DLCEP immediately after transmission of the EC DLPDU.

5) The DLE shall cancel the user-request timer $T_U(MCD)$.

6) If (e) did not apply, then the DLE shall activate recognition of the DLCEP's local DLCEP-address and change the DLCEP state, $V_C(ST)$, to DATA-TRANSFER -READY.

9.2.1.3 Receipt of a DL-CONNECT response primitive

When the DLE receives a DL-CONNECT response primitive from a DLS-user, the DLE shall perform the following series of actions; if any error is detected during the process, then the DLCEP shall be disconnected as specified in 9.2.1.8:

a) The DLE shall validate the DLCEP-identifier, and the responding DLSAP-address or DLCEP-identifier, provided by the DLS-user, and shall associate the provided DLS-user-identifier with the DLCEP.

b) If the identified DLCEP is not in the WAITING-FOR-CONNECT-RESPONSE state, the DLCEP shall be disconnected.

c) The DLE shall validate the self-consistency of the response QoS parameter set, where all static and non-specified parameters assume their default values associated with the responding-DLSAP-address, and where the automatic adjustments to that QoS specified in 9.2.1.2(d) occur. The DLE shall then validate the consistency of the resulting QoS parameter set with the corresponding parameters from the received EC DLPDU, and the adherence to the rules of parameter negotiation specified in 9.2.1.1.

d) If the responding address identifier in the DL-CONNECT response was for a DLCEP, then the DLE shall consider the new DL-CONNECT response as requesting a merger of the response's attempted DLC with an existing DLC, after which the DLCEP-identifier which was specified in the associated DL-CONNECT indication will no longer be valid and any DLS-provider state information (including publisher-DLCEP-address reuse-discriminator) related to the indication's attempted DLC shall be discarded:

1) If the existing DLC is a multi-peer DLC, then the DLE shall consider the DL-CONNECT response to be a request for adding a new subscriber to the existing DLC:

i) If the specified DLCEP is a publisher DLCEP, then the DLE shall

A) use the specified DLCEP's local DLCEP-address and DLSAP-address as the responding DLCEP-address and DLSAP-address for this DL-CONNECT response;

B) set each QoS parameter, and the publisher-DLCEP-address reuse-discriminator, equal to the corresponding parameter of the specified DLCEP, if permitted by the negotiation rules of 9.2.1.1; and

C) if necessitated by the rule of 9.2.1.1(d), then change the address size of the existing DLC from VERY-SHORT to SHORT or from SHORT to LONG.

ii) If no negotiation-rule violation is detected, then the DLE shall

A) encode an EC DLPDU not requesting a reply, with three addresses as specified in 7.1 and 8.1, which are, respectively,

—the first of the two source DL-addresses from the received EC DLPDU which resulted in the DL-CONNECT indication and its consequent DL-CONNECT response;
—the DLCEP-address of the existing DLC; and
—the DLSAP-address associated with this responding DLCEP-address, respectively; and

B) queue the DLPDU at TIME-AVAILABLE priority as specified in 9.4.5; and

C) stop the timer which was started in 9.2.1.4.2(b.4.iv).

iii) Otherwise, when (i) does not apply because the specified DLCEP is a subscriber DLCEP, or when (ii) does not apply because a negotiation rule violation was detected, then the DLE shall reject the received DLC-establishment request and terminate processing of the received EC DLPDU, as follows:

If the destination DL-address of the received EC DLPDU was not a group DL-address, then

A) The DLE shall encode a DC DLPDU as specified in 7.2 and 8.2, with a reason of "connection rejection — QoS not available, permanent condition," and schedule the DLPDU for transmission at TIME-AVAILABLE priority as specified in 9.4.5.

B) The DC DLPDU shall have both destination and source addresses (see 7.2.1 formats 1L and 1S), the destination address shall be identical to the first source DLCEP-address of the received EC DLPDU, which resulted in the DL-CONNECT indication and its consequent DL-CONNECT response, and the source address shall be identical to the destination DL-address of that received EC DLPDU.

2) If the existing DLCEP's DLCEP-class is PEER, then the DLE shall consider the DL-CONNECT response to be a request for resolving a DL-CONNECT request collision on a peer DLC which is in the WAITING-FOR-EC-DLPDU state. The DLE shall apply the negotiation rules of 9.2.1.1 jointly to the QoS parameters of the two DLCEPs and shall reflect the results in the QoS of the originally-requested DLCEP.

If a negotiation rule violation is detected, or if the DLCEP specified by the responding-address is not in the WAITING-FOR-EC-DLPDU state, then both DLCEPs shall be disconnected as specified in 9.2.1.8, but with a reason of "connection rejection — QoS not available, permanent condition."

Otherwise, when no negotiation-rule violation was detected, then the DLE shall

i) use the source DLCEP-address and DLSAP-address, of the received EC DLPDU, which resulted in the DL-CONNECT indication and its consequent DL-CONNECT response, as the remote DLCEP-address and DLSAP-address of the DLCEP which was in the WAITING-FOR-EC-DLPDU state;

ii) consider the reply-requested field of the received EC DLPDU, which resulted in the DL-CONNECT indication and its consequent DL-CONNECT response, to have been set; and

iii) cause the DLCEP specified as the responding-address to

— send a three-address EC DLPDU specifying that a reply is not requested;

— stop the timer which was started in 9.2.1.4.2(b.4.iv);

— start a timer as in 9.2.1.2(b), with a duration equal to the value for the maximum confirm delay on DL-CONNECT as specified in the DL-CONNECT response primitive;

— activate recognition of the DLCEP's local DLCEP-address; and

— change the specified DLCEP's state, $V_C(ST)$, to WAITING-FOR-CONNECT-COMPLETION.

e) If the responding address identifier in the DL-CONNECT response was a DLSAP-address, then

— that DLSAP-address shall be used as the local DLSAP-address; and

— the DLE shall determine the local maximum send and receive window sizes based on the respective buffer-and-queue bindings, possibly further restricted by DL-management, as specified in 9.2.1.2(e.4) through 9.2.1.2(e.7).

The DLE shall then determine the actual maximum send window size as the smaller of the local send window size and the received EC DLPDU's receive window size, and the actual maximum receive window size as the smaller of the local receive window size and the received EC DLPDU's send window size, as specified in 9.2.1.1. The DLE also shall perform all other required negotiations, as specified in 9.2.1.1.

f) If the optional DLCEP-address was specified in the response primitive, then the DLE shall assign that DLCEP-address to the DLCEP; if any conflicting assignment is detected, then the DLCEP shall be disconnected as specified in 9.2.1.8 with a reason of "disconnection — incorrect DLCEP pairing, permanent condition.” Otherwise a DLCEP-address not currently in use shall be assigned to the DLCEP; where possible this shall be a SHORT DL-address of the local link.

When the DLE is serving only as a subscriber in the DLC, then no reply EC DLPDU is permitted, and so no assignment of a DLCEP-address is required.

NOTE — After DLCEP establishment is completed, a subscriber substitutes its calling-DLSAP-address where a sending DLCEP-address otherwise would be required in a CA, CD, ED or DT DLPDU.

g) If the responding DLCEP class is SUBSCRIBER, then the DLE shall

— accept the next received sequence number of the DLC as the sequence number of the first DLSDU;

— stop the timer which was started in 9.2.1.4.2(b.4.iv);

— issue a DL-CONNECTION-ESTABLISHED indication;

— activate recognition of the DLCEP's remote (publisher) DLCEP-address; and

— change the DLCEP's state, $V_C(ST)$, to DATA-TRANSFER-READY.

h) Otherwise, if (g) does not apply, then the DLE shall

1) encode an EC DLPDU not requesting a reply, with three addresses as specified in 7.1 and 8.1, where its addresses are, respectively,

— the first of the two source DL-addresses from the received EC DLPDU which resulted in the DL-CONNECT indication and its consequent DL-CONNECT response;

— the DLCEP-address just assigned to the DLCEP; and

— the responding DLSAP-address, respectively; and

2) assign a publisher-DLCEP-address reuse-discriminator when the responding DLCEP-class is PUBLISHER; and

3) schedule the DLPDU for transmission at TIME-AVAILABLE priority as specified in 9.4.5.

NOTE — This procedure causes a publisher DLCEP to reply to an EC DLPDU from a proposed subscriber DLCEP by sending an EC DLPDU directly to that subscriber DLCEP's temporary calling DLCEP-address. Alternatively, the publishing DLE can create an independent publisher DLCEP (through use of a DL-CONNECT request) after receiving the DL-CONNECT indication from the proposed subscriber, and then merge the subscriber's requested publishing DLCEP into the one just established. This latter approach will cause an EC DLPDU to be sent to the publisher's Called DL(SAP)-address before the response EC DLPDU is sent to the requesting subscriber.

i) If the responding DLCEP class is PUBLISHER, then the DLE shall

— stop the timer which was started in 9.2.1.4.2(b.4.iv);

— issue a DL-CONNECTION-ESTABLISHED indication;

— activate recognition of the DLCEP's local DLCEP-address; and

— change the DLCEP's state, V_C(ST), to DATA-TRANSFER-READY.

j) If the responding DLCEP class is PEER, then the DLE shall

— stop the timer which was started in 9.2.1.4.2(b.4.iv);

— start a timer as in 9.2.1.2(b), with a duration equal to the value for the maximum confirm delay on DL-CONNECT as specified in the DL-CONNECT response primitive;

— activate recognition of the DLCEP's local DLCEP-address; and

— change the DLCEP's state, V_C(ST), to WAITING-FOR-CONNECT-COMPLETION.

9.2.1.4 Receipt of an EC DLPDU

When the DLE receives an EC DLPDU, the DLE shall determine the version number of the DL-protocol in use, as specified in the received EC DLPDU, and shall interpret the other EC-parameters of the DLPDU accordingly.

9.2.1.4.1 Receipt of an EC DLPDU with two addresses

The DLE shall perform the following series of actions, and if any error is detected during the process, then the DLC shall be disconnected as specified in 9.2.1.8:

a) If the first (source) address of the received EC DLPDU is a PUBLISHER DLCEP-address for an existing SUBSCRIBER DLCEP, and if that SUBSCRIBER DLCEP is in the WAITING-FOR-EC-DLPDU state, then the DLE shall

i) validate the received DLC parameters against those earlier requested for the SUBSCRIBER DLCEP (and possibly not sent in an EC DLPDU); and

ii) set the parameters of the existing DLCEP to equal the received DLC parameters, if permitted by the negotiation rules of 9.2.1.1.

If an error is detected, then the DLE shall disconnect the SUBSCRIBER DLCEP as specified in 9.2.1.8, with a reason of "connection rejection — QoS not available, permanent condition."

If no error is detected during this validation, then the DLE shall

iii) set the publisher-DLCEP-address reuse-discriminator equal to that in the received EC DLPDU;

iv) cancel the user-request timer T_U(MCD);

v) accept the next received sequence number of the DLC as the sequence number of the first DLSDU;

vi) issue a DL-CONNECT confirm to the DLS-user; and

vii) change the SUBSCRIBER DLCEP to the DATA-TRANSFER-READY state.

b) When the first (source) address of the received EC DLPDU is a PUBLISHER DLCEP-address for an existing SUBSCRIBER DLCEP, but (a) does not apply because the SUBSCRIBER DLCEP is not in the WAITING-FOR-EC-DLPDU state, then the DLE shall compare the publisher-DLCEP-address reuse-discriminator of the existing DLCEP with that in the received EC DLPDU. If the two values are equal then the DLE shall set the address size of the existing DLCEP to that in the received EC DLPDU.

c) Otherwise, when (a) and (b) do not apply, then the existing SUBSCRIBER DLCEP shall be disconnected as specified in 9.2.1.8, with a reason of "disconnection — wrong publisher-DLCEP-address reuse-discriminator, permanent condition."

9.2.1.4.2 Receipt of an EC DLPDU with three addresses

The DLE shall perform the following series of actions, and if any error is detected during the process, then the DLC shall be disconnected as specified in 9.2.1.8:

a) If the first address in the received EC DLPDU is a group-DL-address associated with more than one of the DLE’s DLS-users, then the DLE shall treat each of those DLS-users as if that user had individually received the EC DLPDU. However, no DC DLPDU shall be sent as a direct response to the received EC DLPDU.

b) If the first address of the received EC DLPDU is a DLSAP-address, then

1) The DLE shall validate the self-consistency of the received EC DLPDU, where all static and non-specified dynamic parameters assume the default values associated with that called-DLSAP-address, and where any parameter in violation of a local DL-management-imposed limit shall be set equal to that limit, if permitted by the negotiation rules of 9.2.1.1, or the DLCEP shall be disconnected as specified in 9.2.1.8 with a reason of "connection rejection — QoS not available, permanent condition.”

2) The DLE shall check whether a DLS-user associated with that DLSAP-address has an active DLCEP whose remote DLCEP-address equals the source DLCEP-address specified in the received EC DLPDU.

3) If such an active DLCEP exists, then

i) if

— the remote DLSAP-address equals the source DLSAP-address specified in the received EC DLPDU;

— the DLCEP is in the WAITING-FOR-CONNECT-COMPLETION state; and

— the received EC DLPDU requests a reply

then a return EC DLPDU, addressed to the first source DL-address specified in the received EC DLPDU, with source addresses equal to the DLCEP’s local DLCEP- and DLSAP-addresses, and specifying the parameters of the active DLCEP, and not requesting a reply, shall be encoded and shall be queued at TIME-AVAILABLE priority as specified in 9.4.5;

ii) else when (i) does not apply, then the DLE shall disconnect the DLCEP as specified in 9.2.1.8 with a reason of "connection rejection — inconsistent DLCEP state, permanent condition;”

4) If no such active DLCEP exists, then the DLE shall assign a new DLCEP identifier to the DLCEP, and shall apply the negotiation rules of 9.2.1.2(d). If any violation of the negotiation rules occurs, then the DLE shall disconnect the proposed DLCEP as specified in 9.2.1.8 with a reason of "connection rejection — QoS not available, permanent condition." If no violation is detected, then for each DLS-user associated with the DLSAP-address or group DL-address which was the first address of the received EC DLPDU, the DLE shall

i) create a DLCEP, initializing its $V_S(NP)$, $V_S(N)$, $V_S(R)$, $V_S(A)$, $V_S(M)$, $V_S(MS)$, $V_S(H)$, $V_S(HS)$ and $V_S(L)$ variables as specified in 5.7.4;

ii) record the source DLCEP-address and source DLSAP-address from the received EC DLPDU as the DLCEP's remote DLCEP-address and remote DLSAP-address, respectively and when sender's DLCEP class is PUBLISHER, also record the publisher-DLCEP-address reuse-discriminator of the EC DLPDU as the DLCEP's local publisher-DLCEP-address reuse-discriminator;

iii) report a DL-CONNECT indication to the DLS-user;

iv) start a timer to monitor for the DLS-user's response to the DL-CONNECT indication, as specified in 9.2.1.2(b); and

v) change the DLCEP state, $V_C(ST)$, to WAITING-FOR-CONNECT-RESPONSE.

c) Else if the first address of the received EC DLPDU is a DLCEP-address for an existing DLCEP, and if the addressed DLCEP is in the WAITING-FOR-EC-DLPDU state, then the DLE shall validate the received DLC parameters, and the received DLPDU's source DL-addresses when their expected values are known, against those sent in an earlier EC DLPDU, and if an error is detected, then

1) If the called address of the DLCEP was a group DL-address, then the DLE shall reply with a DC DLPDU not requesting a reply, addressed to the first source DL-address specified in the received EC DLPDU, with a source address equal to the DLCEP's local DLCEP-address, and shall otherwise ignore the received EC DLPDU.

2) Otherwise, when the called address of the DLCEP was a DLSAP-address, then the DLE shall disconnect the DLCEP as specified in 9.2.1.8 with a reason of "connection rejection — QoS not available, permanent condition."

If no error is detected during the validation of the received EC DLPDU, then

3) If the receiving DLCEP's DLCEP-class is PEER and the received EC DLPDU requests a reply, then

i) the two source DL-addresses of the received EC DLPDU shall be noted as the remote-DLCEP-address and remote-DLSAP-address of the DLCEP;

ii) a return EC DLPDU, addressed to the first source DL-address specified in the received EC DLPDU, with source addresses equal to the DLCEP's local DLCEP- and DLSAP-addresses, and specifying the parameters of the active DLCEP, and not requesting a reply, shall be encoded and shall be queued at TIME-AVAILABLE priority as specified in 9.4.5;

iii) the DLE shall start a timer as specified in 9.2.1.2(b) with a duration equal to the value for the maximum confirm delay on DL-CONNECT as specified in the DL-CONNECT request primitive; and

iv) the DLE shall change the state to WAITING-FOR-CONNECT-COMPLETION.

4) Else when the receiving DLCEP's DLCEP-class is SUBSCRIBER or the received EC DLPDU does not request a reply, then

i) if the receiving DLCEP's DLCEP-class is PEER or SUBSCRIBER, then the two source DL-addresses of the received EC DLPDU shall be noted as the remote-DLCEP-address and remote-DLSAP-address of the DLCEP;

ii) if the DLCEP-class of the receiving DLCEP is PEER, then

— a DT DLPDU not containing DLS-user-data;

— with a destination address equal to the first source DL-address specified in the received EC DLPDU; and

— when the DLCEP's attributes require the DLPDU to have a source address, with a source address equal to the DLCEP's local DLCEP-address

shall be encoded and shall be queued at the DLCEP's priority as specified in 9.4.5, to notify the peer DLE of the successful receipt of the confirming EC DLPDU

iii) if the receiving DLCEP's DLCEP-class is SUBSCRIBER then the DLE shall accept the next received sequence number of the DLC as the sequence number of the first DLSDU;

iv) the DLE shall issue a DL-CONNECT confirm primitive, conveying the negotiated DLCEP-attributes, to the requesting DLS-user; and

v) the DLE shall cancel the user-request timer T_U(MCD) and change the DLCEP state to DATA-TRANSFER-READY.

d) Else if the first address of the received EC DLPDU is a DLCEP-address for an existing DLCEP, and if the addressed DLCEP is in the WAITING-FOR-CONNECT-COMPLETION state, then the DLE shall validate the received DLC parameters against those sent in an earlier EC DLPDU, and if an error is detected, then the DLE shall disconnect the DLCEP as specified in 9.2.1.8 with a reason of "connection rejection — QoS not available, permanent condition."

If no error is detected during this validation, then

1) If the received EC DLPDU requests a reply, then

i) a return EC DLPDU, addressed to the first source DL-address specified in the received EC DLPDU, with source addresses equal to the DLCEP's local DLCEP- and DLSAP-addresses, and specifying the parameters of the active DLCEP, and not requesting a reply, shall be encoded and shall be queued at TIME-AVAILABLE priority as specified in 9.4.5; and

ii) the DLE shall restart the timer with the same period as the previous time.

2) Else when (1) does not apply, then

i) a DT DLPDU not containing DLS-user-data,

— with a destination address equal to the first source DL-address specified in the received EC DLPDU; and

— if the DLCEP's attributes require the DLPDU to have a source address, with a source address equal to the DLCEP's local DLCEP-address

shall be encoded and shall be queued at the DLCEP's priority as specified in 9.4.5, to notify the peer DLE of the successful receipt of the confirming EC DLPDU;

ii) if the state before WAITING-FOR-CONNECT-COMPLETION was WAITING-FOR-EC-DLPDU, then the DLE shall issue a DL-CONNECT confirm primitive, conveying the negotiated DLCEP-attributes, to the requesting DLS-user;

iii) else, when (ii) does not apply, then the state before WAITING-FOR-CONNECT-COMPLETION was WAITING-FOR-CONNECT-RESPONSE, and the DLE shall issue a DL-CONNECTION-ESTABLISHED indication primitive to the responding DLS-user;

iv) the DLE shall cancel the user-request timer T_U(MCD) and change the DLCEP state to DATA-TRANSFER-READY.

e) Else if the first address of the received EC DLPDU is a DLCEP-address for an existing DLCEP, and the received EC DLPDU requests a reply, and if the addressed DLCEP is in the DATA-TRANSFER-READY state, then

1) If the existing DLCEP is a publisher DLCEP, then the DLE shall

i) set each QoS parameter, and the publisher-DLCEP-address reuse-discriminator, equal to the corresponding parameter of the specified DLCEP, if permitted by the negotiation rules of 9.2.1.1; and

ii) if necessitated by the rule of 9.2.1.1(d), then change the address size of the existing DLC from VERY-SHORT to SHORT or from SHORT to LONG.

2) If no negotiation-rule violation is detected, then the DLE shall

i) encode an EC DLPDU not requesting a reply, with two addresses as specified in 7.1 and 8.1, where its addresses are, respectively,

— the DLCEP-address of the existing DLC; and

— the DLSAP-address associated with this existing DLCEP-address, respectively; and

i) schedule the EC DLPDU for transmission at TIME-AVAILABLE priority as specified in 9.4.5.

3) When (2) does not apply, because a negotiation rule violation was detected, then the DLE shall reject the received DLC-establishment request and terminate processing of the received EC DLPDU, as follows:

i) The DLE shall encode a DC DLPDU as specified in 7.2 and 8.2, with its reply-requested field set to FALSE, with a reason of "provider-originated disconnection — QoS not available, permanent condition," and schedule the DC DLPDU for transmission at TIME-AVAILABLE priority as specified in 9.4.5; and

ii) The DC DLPDU shall have both destination and source addresses (see 7.2.1 formats 1L and 1S), the destination address shall be identical to the first source DL-address of the received EC DLPDU, and the source address shall be identical to the destination DL-address of that received EC DLPDU.

f) Otherwise, the DLE shall ignore the received EC DLPDU.

9.2.1.5 Expiration of the timer T_U(MCD)

If the timer T_U(MCD) expires, then if the DLCEP state, V_C(ST), is

a) WAITING-FOR-EC-DLPDU, then

1) if this is the (V(NRC)+1)'th consecutive expiration, then

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

i) the DLE shall terminate processing of the request; and

ii) if the user-specified maximum confirm delay on the DL-CONNECT request primitive specified a value other than UNLIMITED, then

A) the DLE shall initiate a DL-DISCONNECT indication reporting "connection rejection — DLSAP unreachable, transient condition, local origin;" and

B) if the called address was either a DLSAP-address or a DLCEP-address, and the DLCEP's DLCEP-class is PEER, then the DLE

—shall encode a DC DLPDU requesting disconnect, with a reason of "reason unspecified," to the same DL-address as that to which the previous EC DLPDU had been sent; and

— shall be queued at TIME-AVAILABLE priority as specified in 9.4.5;

2) otherwise, if (1) does not apply, then the DLE shall

i) restart the timer with the same period as the previous time; and

ii) requeue the same EC DLPDU for retransmission at TIME-AVAILABLE priority as specified in 9.4.5.

b) WAITING-FOR-CONNECT-RESPONSE, then the DLE shall disconnect the DLCEP as specified in 9.2.1.8, specifying a disconnect reason of "provider-originated disconnection — timeout."

c) WAITING-FOR-CONNECT-COMPLETION, then

1) if this is the (V(NRC)+1)'th consecutive expiration, then the DLE shall disconnect the DLCEP as specified in 9.2.1.8, specifying a disconnect reason of "provider-originated disconnection — timeout;"

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

2) otherwise, if (1) does not apply, then the DLE shall resend the three-address EC DLPDU which was sent when the WAITING-FOR-CONNECT-COMPLETION state was entered, and again restart the timer as specified in 9.2.1.2(b).

d) DATA-TRANSFER-READY, then the DLE shall act as specified in 9.2.2.10.

e) WAITING-FOR-RESET-COMPLETION, then the DLE shall act as specified in 9.2.2.18.

9.2.1.6 Receipt of a DL-DISCONNECT request primitive

When the DLE receives at a DLCEP a DL-DISCONNECT request from a DLS-user, then the DLE

a) shall encode a DC DLPDU as specified in 7.2 and 8.2, requesting disconnect and specifying the DLS-user-given reason, and shall schedule the DLPDU for transmission at TIME-AVAILABLE priority as specified in 9.4.5, except when the DLCEP:

1) is in the WAITING-FOR-CONNECT-RESPONSE state, and the destination DL-address of the EC DLPDU which activated the DLCEP is a group DL-address; or

2) is in the WAITING-FOR-EC-DLPDU state, and the called DL(SAP)-address is a group DL-address or is UNKNOWN; or

3) is a SUBSCRIBER DLCEP;

If a DC DLPDU is encoded, then

i) If the DLCEP being disconnected is a PEER DLCEP, then the DC DLPDU shall have both destination and source addresses (see 7.2.1 formats 1L and 1S), and the destination address shall be the remote DLCEP-address of the DLC, if known, or the called-DL(SAP)-address of the initiating EC DLPDU in all other cases. The reply-requested field shall be set to TRUE in the DC-parameters of the initiating DC DLPDU.

ii) If the DLCEP being disconnected is a PUBLISHER DLCEP, then the DC DLPDU shall have only a source address (see 7.2.1 formats 2L and 2S). The reply-requested field shall be set to FALSE in the DC-parameters of the initiating DC DLPDU.

iii) The source address of the DC DLPDU shall be the local DLCEP-address, if one exists; or the responding or calling local DLSAP-address, if one exists, or the called-DLSAP-address of the initiating EC DLPDU in all other cases.

b) shall terminate the DLCEP, including

1) for each outstanding (that is, not-yet-confirmed) DL-DATA request:

i) remove the request from the appropriate DLCEP user-request queue, Q_A(UR), and references to the request from all DLE queues;

ii) initiate a DL-DATA confirm with the associated request identifier reporting "failure — reset or disconnection;" and

iii) delete the timer T_U(MCD) associated with the request.

2) if a DL-RESET request is outstanding, then initiate a DL-RESET confirm with the DLS-user's DLCEP-identifier reporting "failure — disconnection;" and

3) delete all timers associated with the DLCEP.

After disconnection, if a local DLCEP-address associated with the DLCEP was included in any DLPDU sent on the link, then the DLE shall ensure that the DLCEP-address is not reused for a period of time exceeding the greater of

1) twice the maximum DLPDU lifetime in the network, V(NDL); and

2) the DLCEP's D…D_{CR} delay, when that delay is not UNLIMITED.

9.2.1.7 Receipt of a DC DLPDU

When the DLE receives a DC DLPDU, specifying that the DLCEP should be disconnected, then

a) the DLE shall determine the version number of the DL-protocol in use, as specified in the received DC DLPDU, and shall interpret the other DC-parameters of the DLPDU accordingly;

b) if the received DC DLPDU requests a reply, then a return DC DLPDU, addressed to the source DL-address specified in the received DC DLPDU, and specifying a disconnect reason of "disconnection or connection rejection, unknown origin — reason unspecified," and not requesting a reply, shall be encoded and shall be queued at TIME-AVAILABLE priority as specified in 9.4.5; and

c) if the received DC DLPDU

1) specifies only a source address (see 7.2.1 formats 2L and 2S) and the source address is a DLCEP-address of a multi-peer DLC to which the DLE is a subscriber; or

2) specifies both destination and source addresses (see 7.2.1 formats 1L and 1S); and

i) the destination address is a DL(SAP)-address, and the DLE has a DLCEP, at a DLSAP to which that DL(SAP)-address is bound, whose remote DLCEP-address has the same value as the received source DL-address; or

ii) the destination address is a DLCEP-address, and the remote DLCEP-address of the identified DLCEP has the same value as the received source DL-address; or

iii) the destination address is a DLCEP-address, and the called DLSAP-address of the identified DLCEP has the same value as the received source DL-address

then if the DLCEP is known to the local DLS-user, then

3) the DLE shall report a DL-DISCONNECT indication to the local DLS-user specifying both the non-local origin and the reason for the DL-DISCONNECT indication as received in the DC DLPDU;

4) the DLE shall terminate the DLCEP as specified in 9.2.1.6(b); and

5) after disconnection, the DLE shall ensure that any DLCEP-address which had been assigned to the DLCEP is not reused for a period of time exceeding the greater of

i) twice the maximum DLPDU lifetime in the network, V(NDL); and

ii) the DLCEP's D…D_{CR} delay, when that delay is not UNLIMITED.

9.2.1.8 DLE-initiated disconnection

When the DLE determines on its own that it is necessary to disconnect the DLCEP, then

a) If the DLCEP is known to the local DLS-user, then the DLE shall report a DL-DISCONNECT indication to the local DLS-user, specifying both the reason for the DL-DISCONNECT indication and that its origin was local.

NOTE — The DLCEP will not be known to the local DLS-user if the disconnection occurs while processing a received EC DLPDU whose receipt had just triggered the DL to create the DLCEP.

b) If

1) the DLCEP's DLCEP-class is PEER or PUBLISHER; and

2) the called DL(SAP)-address of the EC DLPDU which activated the DLCEP was not a group DL-address,

then

i) the DLE shall encode a DC DLPDU as specified in 7.2 and 8.2, and shall schedule the DLPDU for transmission at TIME-AVAILABLE priority as specified in 9.4.5;

ii) if the DLCEP being disconnected is a PEER DLCEP, then the DC DLPDU shall have both destination and source addresses (see 7.2.1 formats 1L and 1S), and the destination address shall be the remote DLCEP-address of the DLC, if known, or the called-DL(SAP)-address of the initiating EC DLPDU in all other cases. The reply-requested field shall be set to TRUE in the DC-parameters of the initiating DC DLPDU;

iii) if the DLCEP being disconnected is a PUBLISHER DLCEP, then the DC DLPDU shall have only a source address (see 7.2.1 formats 2L and 2S) and the reply-requested field shall be set to FALSE in the DC-parameters of the initiating DC DLPDU; and

iv) the source address of the DC DLPDU shall be the local DLCEP-address, if one exists; or the responding or calling local DLSAP-address, if one exists, or the called-DLSAP-address of the initiating EC DLPDU in all other cases.

c) the DLE shall terminate the DLCEP as specified in 9.2.1.6(b).

After disconnection, the DLE shall ensure that any DLCEP-address which had been assigned to the DLCEP is not reused for a period of time exceeding the greater of

1) twice the maximum DLPDU lifetime in the network, V(NDL); and

2) the DLCEP's D…D_{CR} delay, when that delay is not UNLIMITED.

9.2.2 Operation of the DLC data transfer and DLCEP reset services

The primitives of the DLC data transfer service are DL-DATA request, indication and confirm, DL-BUFFER-SENT indication, and DL-BUFFER-RECEIVED indication. The primitives of the DLCEP reset services are DL-RESET request, indication, response and confirm, and DL-RESET-COMPLETED indication. A DLCEP reset may be invoked, by any DLS-user or the DLS-provider, at any time after DLCEP establishment and before DLCEP release.

If the sending DLCEP data delivery features in either direction of the DLC are ORDERED, DISORDERED, or CLASSICAL, then the linear (conceptually unlimited) DLCEP variables specified in 5.7.4 and Figure 3 are projected onto a cyclic ring of 2^N sequence identifiers as shown in Figure 8. Each DLSDU is associated with one of these identifiers prior to transmission, and all DLPDUs which convey segments of the DLSDU also specify that associated identifier. The dynamic partitioning of these identifiers, and the resultant categorization of DLSDUs associated with these identifiers, is also shown in Figure 8.

NOTE — The relationships among the linear (conceptually unlimited) DLCEP variables specified in 5.7.4 and Figure 3 are repeated here to assist in the understanding of Figure 8.

$H–W \le M–W \le A \le L \le H \le M \le A+W \le L+W$

$M–W \le R \le M$

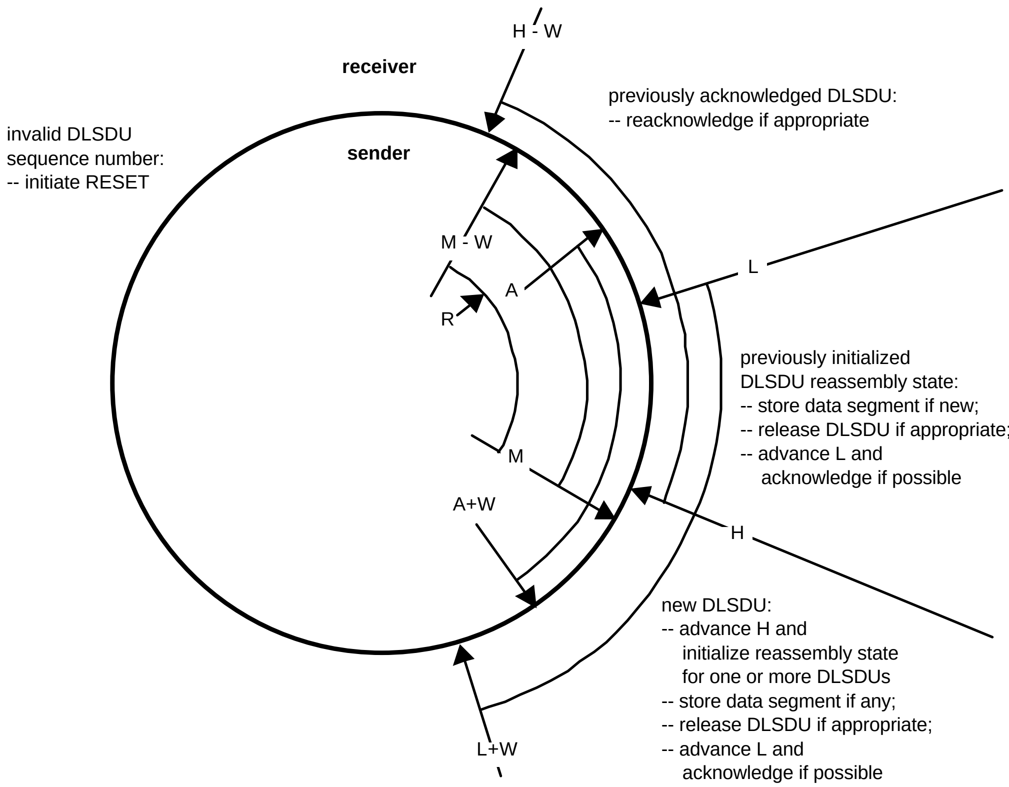


Figure 8 — Projection of the sending and receiving DLCEP sequence-number variables of Figure 3 onto the cyclic sequence-number parameters of CA, CD, DT, ED and RC DLPDUs, with consequent determination of required actions

Each DLPDU with non-null SD-parameters can convey to the peer DLE the low-order 2 to 5 bits of the sender's $V_C(L)$ and $V_C(M)$, as the (NDR mod 2^R) and (NDS mod 2^S) subfields of the SD-parameters.

9.2.2.1 Selection of the format of a CA, CD, DT and ED DLPDUs

The address format of all CA, CD, ED, and DT DLPDUs sent from a DLCEP shall be chosen as determined during the DLCEP-establishment process (see 9.2.1.1) and as specified in 7.4.3, 7.5.3, 7.6.3 and 7.7.3, respectively. The SD-parameter format of all such CA, CD, ED and DT DLPDUs formed by the DLE shall be the same as that negotiated for the sending DLCEP (see 8.1(c.5), 8.1(d.5), and 9.2.1.1).

All CD DLPDUs sent from an LAS DLE as part of its schedule execution activities, and not from a DLCEP of the LAS DLE, shall specify an explicit destination address of the length negotiated in 9.2.1.1 and shall omit both source address and SD-parameters.

NOTE — An address format of VERY-SHORT is always realized by use of SHORT addresses in any associated CA, CD, DC, ED and RC DLPDUs.

9.2.2.2 Receipt of a DL-DATA request primitive

If the request is accepted, as indicated by a returned status of "success" for the DL-DATA request, then upon completion of the request, either successfully or after failure, the DLE shall issue a DL-DATA confirm with the same request identifier as specified by the DLS-user in the corresponding DL-DATA request primitive, conveying the status of the request to the DLS-user.

The DLCEP source specified in the DL-DATA request should be bound to either an explicit (user-controlled) queue or to an implicit (DLE-controlled) queue. If the queue is full, or if the specified DLSDU length, $P_U(L)$, is invalid, or if the DLCEP-state $V_C(ST)$ is not DATA-TRANSFER-READY, then the DLE shall immediately return the corresponding DL-DATA confirm indicating the reason for failure.

Otherwise

a) The DLE shall create and start a user-request timer $T_U(MCD)$ with a duration based on the user-specified maximum confirm delay for DL-DATA primitives. If the specified value was other than UNLIMITED, then the duration of this timer shall be equal to that user-specified maximum confirm delay; otherwise the duration should be 60 s. DL-management may override these preferred durations.

b) The DLE shall assign the next unassigned sequence number $N = V_C(N)$ to the request and its associated DLSDU.

c) The DLE shall initialize the variable $V_{C,N}(SS)$ based on the length, $P_N(L)$, of the Nth DLSDU, to indicate that all segments of the Nth DLSDU, and no other segments of that DLSDU, need transmission.

d) The DLE shall append the request to the DLCEP-address's user-request queue, $Q_A(UR)$, as follows:

1) If the DL-scheduling-policy attribute of the source DLCEP-address is EXPLICIT, then the request shall be placed in the third partition of $Q_A(UR)$, where it shall await a DL-COMPEL-SERVICE request to release it for transmission.

NOTE — Release of a deferred request applies to DLSDUs, not DLPDUs.

2) Else if (1) does not apply; and

i) if $N > V_C(A) + P_C(WS)$, and the sending DLCEP is a CLASSICAL or DISORDERED peer; or

ii) if $N > V_C(R) + P_C(WS)$, and the sending DLCEP is ORDERED and $V_C(R)$ exists, or is a CLASSICAL or DISORDERED publisher or subscriber;

then the request shall be placed in the third partition of $Q_A(UR)$;

3) Else if (1) and (2) do not apply, then the third partition of $Q_A(UR)$ is empty, and so the request shall be placed in the second partition of $Q_A(UR)$, and the DLE shall append to the DLE's unscheduled-service queue, Q(US), a reference to $Q_A(UR)$ of the same priority as the just-appended request.

NOTE — Q(US) never needs to have more references to a $Q_A(UR)$ than the number of DLSDUs waiting for transmission or retransmission.

e) The DLE shall increment $V_C(N)$.

9.2.2.3 Transmission of a DT DLPDU from a DLCEP

In the following, let MUD be defined as the maximum number of user-data octets which may be conveyed in a single DT DLPDU of the DLCEP's priority, as specified in 5.2.3.

Upon receipt of a transmission opportunity for a queued or buffered DLSDU, the DLE shall form and send a DT DLPDU of the specified priority; with DL-address field and SD-parameter field formats as specified in 7.7, 8.4 and 9.2.2.1; with the remote (destination) and local (source) DLCEP-addresses of the DLC, as appropriate; and with user-data-field length and contents, and SD-parameter-field contents, as follows:

NOTE — With the exception of the final-token-use subfield of the DT DLPDU's first (frame control) octet; and probably of the NDR, RSN, J, K and T subfields, if present, of the SD-parameter format negotiated for transmission from the sending DLCEP, the remainder of the data DT DLPDU may be formed when the request is queued, and need not be formed dynamically at the moment of transmission.

9.2.2.3.1 Formation of the user-data field and related SD-parameter subfields

The T, NDS, TNS, ASN and truncated-DL-time subfields of the SD-parameters field, and the DLPDU's user-data field, shall be formed as follows:

a) If

   i) the sending DLCEP is a DISORDERED or CLASSICAL PUBLISHER DLCEP, or an ORDERED PEER or PUBLISHER DLCEP and $V_C(R)$ exists;

   ii) $V_C(M) = V_C(R)+P_C(WS)$ and $V_C(M) +1 < V_C(N)$ and $V_{C,VC(M)}(SS)$ is empty; and

   iii) a DL-DATA confirm primitive has been issued for the ($V_C(R)$+1)'th DL-DATA request

then the DLE shall

1) cancel the timer associated with the ($V_C(R)$+1)'th DL-DATA request; and

2) increment $V_C(R)$.

b) If the sending DLCEP is bound to a sending queue, and is

   i) a SUBSCRIBER DLCEP, or an UNORDERED PEER or PUBLISHER DLCEP, and there is a smallest K such that $V_C(M) < K \le V_C(N)-1$ and $V_{C,K}(SS)$ is non-empty;

   NOTE — In this case $V_{C,K}(SS)$ can contain only a single element, since such DLSDUs cannot require segmentation.

   ii) a DISORDERED or CLASSICAL PEER DLCEP, and there is a K such that $V_C(A) < K \le \min(V_C(A)+P_C(WS), V_C(N)-1)$ and $V_{C,K}(SS)$ is non-empty;

   iii) a DISORDERED or CLASSICAL PUBLISHER DLCEP, or an ORDERED PEER or PUBLISHER DLCEP and $V_C(R)$ exists; and there is a K such that $V_C(R) < K \le \min(V_C(R)+P_C(WS), V_C(N)-1)$ and $V_{C,K}(SS)$ is non-empty; or

   iv) an ORDERED PEER or PUBLISHER DLCEP and $V_C(R)$ does not exist; and there is a K such that $V_C(M) \le K \le \min(V_C(M)+P_C(WS), V_C(N)-1)$ and $V_{C,K}(SS)$ is non-empty; or

then the DLE shall form the remainder of the DLPDU as follows:

1) the NDS subfield, if present, shall convey the lowest-order three or five bits of the value K, as appropriate;

2) if S is the zero-origin index of the lowest-numbered member of the set $V_{C,K}(SS)$, which is the segment number of a segment of the K'th DLSDU requiring transmission, then the ASN subfield, if present, shall convey the value of S;

NOTE — It is possible for an implementation of this DL-protocol to perform DLSDU segmentation at the time of DLPDU transmission. Such an implementation approach could minimize the DLE's processing of DLS-user-data.

3) the TNS subfield, if present, shall convey the value (P_K(SDUL) -1)/MUD, rounded down, which is the zero-origin number of segments in the K'th DLSDU;

4) the user-data field shall consist of octets

$S \times MUD + 1$ through $\min((S+1) \times MUD, P_K(SDUL))$, inclusive, of P_K(SDU);

5) the T subfield shall specify FALSE; and

6) the truncated-DL-time subfield shall not be present;

and the DLE shall remove member number S from $V_{C,K}(SS)$.

c) If the sending DLCEP is bound to a sending buffer, then

1) the DLE shall increment $V_C(N)$.if the buffer has been written since the last transmission from the buffer on this DLCEP;

2) the DLE shall let K equal $V_C(N)$-1;

3) if K is not equal to zero, then the DLE shall form the remainder of the DLPDU as specified in (b.1) through (b.4) and as follows:

i) if the DLSDU has no timeliness attribute, then the T subfield shall specify FALSE;

ii) if the DLCEP has a sender's-DL-timeliness class other than NONE, then the DLE shall:

A) compute the timeliness of the S'th segment of the K'th DLSDU as specified in 9.1.7;

B) perform a logical AND of that computed timeliness status with the timeliness-status associated with writing the buffer, $V_B(TS)$ (see 5.7.4.21); and

C) convey that result in the T subfield of the DLPDU; and

iii) if present, the truncated-DL-time subfield shall convey the appropriate octets of the time-of-production associated with the buffer;

and the DLE shall remove member number S from $V_{C,K}(SS)$.

d) If there is no such K as in (b), or if K is equal to zero in (c), or if the DLE is required to send the DLPDU without user-data as in 7.4.4.1(c), 7.5.4.1(c), and 7.6.4.1(c); or 7.4.4.1(d), and 7.5.4.1(d), 7.6.4.1(d), then

1) the T, TNS, ASN and truncated-DL-time subfields of the SD-parameters, if present, shall be encoded as zero (0); NDS shall be encoded as the appropriate number of low-order bits of $V_C(M)$;

2) the truncated DL-time subfield may be null (that is, omitted and not transmitted); and

3) the user-data field shall be null.

9.2.2.3.2 Formation of the other SD-parameter subfields

The J, K, NDR and RSN subfields of the SD-parameters shall be formed as follows:

a) If the sending DLCEP is a PUBLISHER DLCEP, then the J, K, NDR and RSN subfields of the SD-parameters shall be encoded as zero (0);

b) If the sending DLCEP is a PEER or SUBSCRIBER DLCEP, then

1) If there is a smallest K such that $V_C(L) < K \leq V_C(H)$ and $V_{C,K}(RRS)$ is non-empty, then if S is the zero-origin index of the lowest-numbered member of the set $V_{C,K}(RRS)$, which is the segment number of a missing segment of the K'th DLSDU, and either $K < V_C(H)$ or $S \leq V_C(HS)$, then

i) the J subfield of the SD-parameters, if present, shall be encoded as one (1);

ii) the NDR subfield, if present, shall convey the lowest-order two or four bits of the value K;

iii) the RSN subfield, if present, shall convey the value of S; and

iv) segment S shall be removed from $V_{C,K}(RRS)$.

2) Otherwise, if (1) does not apply, either the SD-parameter field shall be null, if permitted by 9.2.2.1; or

i) the J and RSN subfields shall be encoded as zero (0), if present; and

ii) the NDR subfield, if present, shall convey the lowest-order two or four bits of $(V_C(L)+1)$.

3) If the K and NDR subfields of the SD-parameters are both present, and the value of the NDR subfield equals the value of the corresponding lowest-order bits of $(V_C(L)+1)$, then the K subfield shall be encoded as one (1); otherwise the K subfield shall be encoded as zero (0).

9.2.2.3.3 Transmission completion

a) If the just-transmitted DLPDU contained DLS-user-data, then the DLE shall update $V_C(M)$ and $V_C(MS)$ from the local variables K and S of 9.2.2.3.1 as follows:

— If $K > V_C(M)$, or $K = V_C(M)$ and $S > V_C(MS)$, then $V_C(M)$ shall be set equal to K, and $V_C(MS)$ shall be set equal to S.

b) If the sending DLCEP

— is a peer DLCEP whose sending DLCEP features are UNORDERED, or is a publisher DLCEP whose sending DLCEP features are UNORDERED, or is a subscriber DLCEP; and

— the DT DLPDU has a non-null user data field which contains the last, or only, segment of a DLSDU;

then

1) if the sending source is a buffer, then the DLE shall issue a DL-BUFFER-SENT indication primitive specifying the DLS-user-identifier if known, or the DL-identifier otherwise, for the DLCEP; or

2) if the sending source is a queue and there is an unconfirmed DL-DATA request for that DLSDU, then the DLE shall

i) issue a DL-DATA confirm primitive with the same request identifier as that DL-DATA request, reporting “success;” and

ii) cancel and delete the associated timer $T_U(MCD)$.

c) If the sending DLCEP

— is a publisher DLCEP whose sending DLCEP features are ORDERED, DISORDERED or CLASSICAL, or is a peer DLCEP whose sending DLCEP features are ORDERED; and

— the DT DLPDU has a non-null user data field which contains the last, or only, segment of a DLSDU;

then

1) if the sending source is a buffer, then the DLE shall issue a DL-BUFFER-SENT indication primitive specifying the DLS-user-identifier if known, or the DL-identifier otherwise, for the DLCEP; or

2) if the sending source is a queue and there is an unconfirmed DL-DATA request for that DLSDU, then

i) the DLE shall issue a DL-DATA confirm primitive with the same request identifier as that DL-DATA request, reporting “success;”

ii) if $V_C(R)$ exists, and $V_C(M) – P_C(WS) > V_C(R)$, then the DLE shall set $V_C(R)$ equal to $V_C(M) – P_C(WS)$; and

iii) the DLE shall retain the DLSDU for potential retransmission to subscribers

— until the expiration of the associated timer $T_U(MCD)$;or

— until the need to transmit another DLSDU with a resultant increase in $V_C(M)$, when coupled with the negotiated window size $P_C(WS)$, requires the discard of the retained DLSDU with a resultant increase in $V_C(R)$.

d) If this DLCEP has been specified as a synchronizing DLCEP during the establishment of one or more other local DLCEPs, and if a DL-BUFFER-SENT indication primitive was issued in (b.1) or (c.1), then the DLE shall record the DL-time of network access, $V_C(TNA)$, for use in the timeliness computations of those referencing DLCEP(s).

9.2.2.4 Transmission of a CA, CD or ED DLPDU from a DLCEP

9.2.2.4.1 Transmission of a CA DLPDU

Upon receipt of a transmission opportunity to compel transmission from a remote DLCEP, when

a) the DLC is

1) simplex, with DLS-user-data transmission only from the local DLCEP to the remote DLCEP; or

2) duplex, with DLS-user-data transmission from the local DLCEP to the remote DLCEP, and from that remote DLCEP to the local DLCEP (and possibly other DLCEPs), and the DLCEP attributes do not permit sending an ED DLPDU from the DLCEP to the remote DLCEP; and

b) the sending DLE needs to determine the state of the remote DLCEP;

then the DLE shall form and send a CA DLPDU of the specified priority; with DL-address field and SD-parameter field formats as specified in 7.5, 8.4 and 9.2.2.2; with the remote (destination) and local (source) DLCEP-addresses of the DLC, as appropriate; and with SD-parameter-field contents and user-data-field contents as specified in 9.2.2.3.

9.2.2.4.2 Transmission of a CD DLPDU

This subclause does not apply to the LAS DLE when it sends CD DLPDUs as part of its scheduled activity and not from a DLCEP of the LAS DLE; such DLPDUs are constrained as specified in 7.5 and 9.2.2.

Upon receipt of a transmission opportunity to compel transmission from a remote DLCEP, when

a) the DLC is

1) simplex, with DLS-user-data transmission only from the remote DLCEP to the local DLCEP (and possibly other DLCEPs); or

2) duplex, with DLS-user-data transmission from the local DLCEP to the remote DLCEP, and from that remote DLCEP to the local DLCEP (and possibly other DLCEPs), and either

i) the DLCEP attributes do not permit sending an ED DLPDU from the local DLCEP to the remote DLCEP; or

ii) there are no DLSDU segments awaiting transmission to the remote DLS-user as specified by the selection criteria of 9.4.2.1(a); and

b) the local execution of a DL-COMPEL-SERVICE request primitive, or of a scheduled compel-service action, compels transmission from a remote peer or publisher DLCEP;

then the DLE shall form and send a CD DLPDU of the specified priority; with DL-address field and SD-parameter field formats as specified in 7.5, 8.4 and 9.2.2.2; with the remote (destination) and local (source) DLCEP-addresses of the DLC, as appropriate; and with SD-parameter-field contents, as follows:

NOTE — With the exception of the final-token-use subfield of the CD DLPDU's first (frame control) octet; and probably of the NDR, RSN, J and K subfields, if present, of the SD-parameter format negotiated for transmission from the sending DLCEP, the remainder of the CD DLPDU may be formed when the request is queued, and need not be formed dynamically at the moment of transmission.

1) the T, NDS, TNS, ASN and truncated-DL-time subfields of the SD-parameters, if present, shall be encoded as zero (0);

2) the truncated DL-time subfield may be null (that is, omitted and not transmitted);

3) the user-data field shall be null; and

4) the J, K, NDR and RSN subfields of the SD-parameters shall be formed as specified in 9.2.2.3.2.

9.2.2.4.3 Transmission of an ED DLPDU

Upon receipt of a transmission opportunity to compel transmission from a remote DLCEP, when

a) the DLC is duplex, with DLS-user-data transmission from the local DLCEP to the remote DLCEP, and from that remote DLCEP to the local DLCEP (and possibly other DLCEPs);

b) the DLCEP attributes permit sending an ED DLPDU from the DLCEP to the remote DLCEP;

c) one or more DLSDU segments await transmission to the remote DLS-user as specified by the selection criteria of 9.4.2.1(a); and

d) the local execution of a DL-COMPEL-SERVICE request primitive, or of a scheduled compel-service action, compels transmission from a remote peer or publisher DLCEP;

then the DLE shall form and send an ED DLPDU of the specified priority; with DL-address field and SD-parameter field formats as specified in 7.5, 8.4 and 9.2.2.2; with the remote (destination) and local (source) DLCEP-addresses of the DLC, as appropriate; and with SD-parameter-field contents and user-data-field contents as specified in 9.2.2.3.

9.2.2.5 Validation and processing of SD-parameters in a CA, CD, ED or DT DLPDU received at a DLCEP

If the DLCEP state, $V_C(ST)$, is

1) WAITING-FOR-CONNECT-COMPLETION, then

i) if the state before WAITING-FOR-CONNECT-COMPLETION was WAITING-FOR-EC-DLPDU, then the DLE shall issue a DL-CONNECT confirm primitive, conveying the negotiated DLCEP-attributes, to the requesting DLS-user and cancel the associated user request timer T_U(MCD);

ii) else, when (i) does not apply, then the state before WAITING-FOR-CONNECT-COMPLETION was WAITING-FOR-CONNECT-RESPONSE, and the DLE shall issue a DL-CONNECTION-ESTABLISHED indication primitive to the receiving DLS-user and cancel the associated user request timer T_U(MCD);

iii) the DLE shall change the DLCEP state, V_C(ST), to DATA TRANSFER READY; and shall apply the remainder of this sub-clause.

2) WAITING-FOR-RESET-COMPLETION, then

i) the DLE shall issue a DL-RESET-COMPLETED indication primitive to the receiving DLS-user specifying the DLS-user-identifier for the DLCEP if known, or the DL-identifier for the DLCEP otherwise;

ii) the DLE shall cancel the associated user request timer T_U(MCD); and

iii) the DLE shall change the DLCEP state, V_C(ST), to DATA-TRANSFER-READY; and shall apply the remainder of this sub-clause.

3) not WAITING-FOR-CONNECT-COMPLETION, and not WAITING-FOR-RESET-COMPLETION, and not DATA-TRANSFER-READY, then the received DLPDU shall be ignored by the upper-level DLC functions.

Otherwise, the DLE shall validate and process the SD-parameters of the received DLPDU according to the SD-parameter format, $P_C(NP.FFF_R)$, negotiated for this (receiving) direction of DLC transmission. This validation and processing shall be as specified in the remainder of 9.2.2.5, with format-dependent considerations as follows, based on the SD-parameter format (A – G)and the truncated DL-time format (J – M). The format-dependent value of the sending modulus MOD_S shall also be used in the procedures of 9.2.2.6:

format A) The sending and receiving SD-parameters of the DLPDU are implicit and thus always valid; the implied values of RSN, T, TNS, ASN and truncated DL-time are all zero; and any accompanying user-data is a complete DLSDU. 9.2.2.5.2 does not apply.

format B) The sending and receiving SD-parameters of the DLPDU are explicit; the sending modulus MOD_S equals 2^3; the receiving modulus MOD_R equals 2^2; the implied values of RSN, TNS and ASN are zero; and any accompanying user-data is a complete DLSDU.

format C) The sending and receiving SD-parameters of the DLPDU are explicit; the sending modulus MOD_S equals 2^5; the receiving modulus MOD_R equals 2^4; and any accompanying user-data may be only a partial DLSDU.

format D) The sending and receiving SD-parameters of the DLPDU are explicit; the sending modulus MOD_S equals 2^5; the receiving modulus MOD_R equals 2^4; the implied values of TNS and ASN are zero; and any accompanying user-data is a complete DLSDU.

format E) The sending SD-parameters of the DLPDU are non-existent; the receiving SD-parameters of the DLPDU are explicit; the receiving modulus MOD_R equals 2^4; the implied values of J, K and T are as specified in 8.4.2(E); the implied value of NDS is $V_C(H)+1$; the implied values of TNS and ASN are zero; and there cannot be accompanying user-data.

format F) The sending SD-parameters of the DLPDU are explicit; the receiving SD-parameters of the DLPDU are non-existent; the sending modulus MOD_S equals 2^5; the implied value of NDR is $V_C(M)+1$; the implied value of RSN is zero; and any accompanying user-data may be only a partial DLSDU.

format G) The sending SD-parameters of the DLPDU are explicit; the receiving SD-parameters of the DLPDU are non-existent; the sending modulus MOD_S equals 2^5; the implied value of NDR is $V_C(M)+1$; the implied values of RSN, TNS and ASN are zero; and any accompanying user-data is a complete DLSDU.

9.2.2.5.1 Validation of the NDS, TNS, ASN and truncated-DL-time subfields of the received SD-parameters

In the following, $P_C(NP.WWWW_R)$ is the negotiated receive window size and $P_C(NP.TT_R)$ is the negotiated receiving DLCEP data delivery features:

a) If $P_C(NP.TT_R)$ specifies UNORDERED, as is always the case with format A, then if the received DLPDU's user-data field is non-null, then the receiving DLE

i) shall increment $V_C(H)$, and shall let *K* equal the new value of $V_C(H)$;

ii) shall create the variable $V_{C,K}(MRS)$ with a value indicating that segment number zero (0) of the *K*'th DLSDU is missing; and

iii) shall process the received user-data as specified in 9.2.2.6.

b) Otherwise, when $P_C(NP.TT_R)$ specifies ORDERED, DISORDERED or CLASSICAL, and if the receiving DLCEP is a subscriber DLCEP, and this is the first DT DLPDU received after the DLCEP state was changed to DATA-TRANSFER-READY, then the DLE shall set the variables $V_C(L)$ and $V_C(H)$ to the value of the $N_R(NDS)$ subfield of the received DT DLPDU.

The DLE shall compute

$$TEMP = \left(N_R(NDS) + P_C(NP.WWWW_R) - V_C(H) - 1 \right) \text{ modulo } MOD_S \qquad \text{(Eq. 15)}$$

1) If

$$TEMP > \left(V_C(L) + 2 \times P_C(NP.WWWW_R) - V_C(H) - 1 \right) \text{ modulo } MOD_S$$

then

i) if $P_C(NP.TT_R)$ is ORDERED, then the DLE shall

A) set $V_C(L)$ equal to

$$V_C(L) + ((N_R(NDS) - (P_C(NP.WWWW_R) + V_C(L))) \text{ modulo } MOD_S);$$

B) cancel all timers $T_{C,N}(RRS)$ which may exist, where *N* is less than or equal to $V_C(L)$. All DLE resources devoted to reception and reassembly of DLSDUs with sequence numbers less than or equal to $V_C(L)$ should be released.

ii) if $P_C(NP.TT_R)$ is CLASSICAL or DISORDERED, then the received DLSDU sequence number is invalid; the procedures of 9.2.2.6 do not apply; and the DLE shall initiate a reset at the DLCEP (see 9.2.2.19).

2) Else if (1.ii) does not apply, then if either

— $TEMP > (P_C(NP.WWWW_R) - 1)$; or

— $TEMP = (P_C(NP.WWWW_R) - 1)$ and $N_R(ASN) > V_C(HS)$;

then

i) The DLE shall set N equal to $TEMP$ – (P_C(NP.WWWW$_R$) –1).

ii) If $N > 0$, then the received DLSDU sequence number is for a new DLSDU, not previously received or inferred; the DLE shall repeat the following step (A) N times, followed by step (B) once:

A) The DLE shall increment V_C(H). Let K equal the just-incremented value of V_C(H). Then $V_{C,K}$(MRS) shall be created and shall indicate that all possible segments of the K'th DLSDU, based on the negotiated maximum-DLSDU-size, are missing; and $V_{C,K}$(RRS) shall be created and shall indicate that segment number zero (0) of the K'th DLSDU is missing.

NOTES —

1.Alternatively, the DLE can simply set $V_{C,K}$(MRS) to indicate that all (16) segments are missing; later procedures will correct the number of missing segments to those of the actual DLSDU.

2.This combination of values for $V_{C,K}$(MRS) and $V_{C,K}$(RRS) ensures that all segments of the K'th DLSDU will be received before the reassembled DLSDU is delivered to the DLS-user.

3.The DLE repeats the above step N times.

B) If the received SD-parameters contain an explicit TNS field, then the DLE shall modify $V_{C,K}$(MRS) to indicate that all segments whose zero-origin number is greater than the value of TNS are not missing, where K equals the new value of V_C(H) after step (A).

iii) If $N = 0$, then the DLE shall set K equal the value of V_C(H).

iv) For all values of N,

A) The DLE shall set the variable V_C(HS) equal to the value of the N_R(ASN) field. If there is any accompanying user data in the received DLPDU, then the DLE shall modify both $V_{C,K}$(MRS) and $V_{C,K}$(RRS) to indicate that the segment whose zero-origin number is equal to the value of N_R(ASN) field is not missing, and the procedures of 9.2.2.6 also shall be applied.

B) If there is any $V_{C,K}$(RRS), as created in (b.2.ii.A), which is not empty and which therefore requires a retransmission request, and if the receiving DLCEP is a CLASSICAL or DISORDERED DLCEP, or optionally is an ORDERED DLCEP, then

—the DLE shall check for a reference to the DLCEP on the DLE's unscheduled-service queue, Q(US); and
—if no such reference is found then the DLE shall add a reference to the DLCEP onto the DLE's unscheduled-service queue, Q(US), to ensure that another DLPDU requesting retransmission of the missing segment, is sent from the receiving DLCEP.

3) Else if (1) and (2) do not apply, and

$$TEMP < (V_C(L) + P_C(NP.WWWW_R) – V_C(H)) \text{ modulo } MOD_S$$

then the received DLSDU sequence number is for a previously delivered, and on peer DLCs previously acknowledged, DLSDU. If there is any accompanying user data in the received DLPDU, and the DLCEP is a CLASSICAL or DISORDERED peer DLCEP, then the DLE shall check for a reference to the DLCEP on the DLE's unscheduled-service queue, Q(US), and if not found then add a reference to the DLCEP to the DLE's unscheduled-service queue, Q(US), to ensure that another DLPDU reacknowledging the just-referenced DLSDU is sent from the receiving DLCEP.

If the DLCEP's receive binding is explicitly or implicitly to a queue, then the procedures of 9.2.2.6 do not apply. If the DLCEP's receive binding is to a buffer, then any DLS-user-data in the DLPDU shall be discarded and the receipt of the duplicate DLPDU shall be reported to the DLS-user with a DL-BUFFER-RECEIVED indication specifying that the reported DLSDU is a duplicate DLSDU.

4) Else if (1), (2) and (3) do not apply, then the received DLSDU sequence number is for a previously received or inferred, but not yet acknowledged, or delivered, or both, DLSDU.

Let $K = V_C(H) + TEMP + 1 - P_C(NP.WWWW_R)$.

If the received SD-parameters contain an explicit TNS subfield, then $V_{C,K}(MRS)$ and $V_{C,K}(RRS)$ both shall be modified to indicate that all segments whose zero-origin number is greater than TNS are not missing. If there is any accompanying user data in the received DLPDU, and $V_{C,K}(MRS)$ indicates that the user data has not previously been received, then the DLE shall modify both $V_{C,K}(MRS)$ and $V_{C,K}(RRS)$ to indicate that the segment whose zero-origin number is equal to the value of $N_R(ASN)$ field is not missing, and the procedures of 9.2.2.6 also shall be applied.

9.2.2.5.2 Validation of the NDR, RSN, J and K subfields of the received SD-parameters

In the following, $P_C(NP.WWWW_R)$ is the negotiated receive window size and $P_C(NP.TT_R)$ is the negotiated receiving DLCEP data delivery features:

a) If the DLCEP is a subscriber DLCEP, and the NDR, RSN, J and K subfields, if present, of the received SD-parameters are not all zero, then the DLE shall disconnect from the DLCEP as specified in 9.2.1.8 with a reason of “provider-originated disconnection — wrong DLPDU format or parameters, permanent condition.”

b) If the DLCEP is a CLASSICAL or DISORDERED peer DLCEP, and the J and K subfields of the received SD-parameters are not both zero, then the DLE shall compute

$$TEMP = (\, N_R(NDR) - V_C(A) \,) \text{ modulo } MOD_R \qquad \text{(Eq. 16)}$$

$$N = TEMP + V_C(A) \qquad \text{(Eq. 17)}$$

The received DLPDU is acknowledging a previously-unacknowledged transmitted DLSDU (K=1), or requesting retransmission of a segment of a previously-transmitted DLSDU (J=1), or both.

If K=1, and the DLCEP is a CLASSICAL or DISORDERED peer DLCEP, and

$V_C(A) < N \le V_C(M) + 1$, then the DLE shall

i) set $V_C(A)$ equal to N-*1*;

ii) issue, in the order originally requested, a DL-DATA confirm for each DL-DATA request which was acknowledged by the received NDR;

iii) cancel the set of associated user request timers { $T_U(MCD)$ } for the just-confirmed DL-DATA requests;

iv) cancel any retransmission timers $T_{C,K}(SS)$ associated with the just-confirmed DL-DATA requests, or the simplified timer $T_C(SS)$ associated with the DLCEP, and in this latter case (using $T_C(SS)$), if $V_C(A) < V_C(M)$, which implies that there are unacknowledged DLSDUs, then $T_C(SS)$ shall be restarted; and

v) where possible and permitted, move DL-DATA requests from the third partition to the second partition of the corresponding user-request queue, $Q_A(UR)$, as specified in 9.2.2.2(d); and if the $V_{C,K}(SS)$ associated with the just-confirmed DL-DATA requests were not empty, then the DLE may cancel such retransmission requests and set the corresponding $V_{C,K}(SS)$ to empty.

If J=1, and N is greater than $V_C(A)$, and either

- $N < V_C(M)$; or
- $N = V_C(M)$ and RSN $\leq V_C(MS)$,

then the DLE shall add the RSN'th member to the set $V_{C,N}(SS)$; and if the set $V_{C,N}(SS)$ was previously empty, then the DLE shall

— cancel any retransmission timers $T_{C,N}(SS)$ associated with the N'th DLSDU, or $T_C(SS)$ associated with the DLCEP; and

— add to the DLE's unscheduled-service queue, Q(US), a reference to $Q_A(UR)$ of the receiving DLCEP, to ensure that the requested DLPDU is sent from the receiving DLCEP.

NOTE — Q(US) never needs to have more references to a $Q_A(UR)$ than the number of DLSDUs waiting for transmission or retransmission.

c) If the DLCEP is an ORDERED DLCEP or a publisher DLCEP, and $V_C(R)$ exists, and the J subfield of the received SD-parameters is not zero, then the DLE shall compute

$$TEMP = (N_R(NDR) - V_C(R)) \text{ modulo } MOD_R \qquad \text{(Eq. 18)}$$

1) If $TEMP > (V_C(M) - V_C(R))$ or,

when $TEMP = (V_C(M) - V_C(R))$ and RSN $> V_C(MS)$,

then the received sequence number residue $N_R(NDR)$ for an acknowledged or a re-requested DLSDU is obsolete or invalid and shall be ignored.

NOTE — Misordering of transmitted DLPDUs, which can result in this condition, is possible when the communications path between the sending and receiving DLEs includes active or backup redundant bridges or DL-paths.

2) Else if (1) does not apply, then the received DLPDU is requesting retransmission of a segment of a previously-transmitted DLSDU (J=1). Let N equal $TEMP + V_C(R)$. If J=1 and N is greater than $V_C(R)$, then the DLE shall add the RSN'th member to the set $V_{C,N}(SS)$; and if the set $V_{C,N}(SS)$ was previously empty, then the DLE shall add to the DLE's unscheduled-service queue, Q(US), a reference to $Q_A(UR)$ of the receiving DLCEP, to ensure that the requested DLPDU is sent from the receiving DLCEP.

NOTE — Q(US) never needs to have more references to a $Q_A(UR)$ than the number of DLSDUs waiting for transmission or retransmission.

9.2.2.5.3 Processing of the T and truncated DL-time subfields of the received SD-parameters

If the DLCEP's receive binding is to a buffer, then

a) If the receiving DLCEP has a sender's DL-timeliness class of NONE, then the timeliness-status, $V_B(TS)$ (see 5.7.4.21), associated with writing the buffer shall be set to FALSE.

b) Otherwise, when (a) does not apply, then

1) If the received DLPDU conveyed the first-received segment of the DLSDU, then

i) the buffer's associated timeliness-status, $V_B(TS)$ (see 5.7.4.21), shall be set equal to the T subfield of the received DLPDU;

ii) if that timeliness-status is TRUE, and if the SD-parameter included time-of-production, format K through M, then the time-of-production of the buffer, $V_B(TP)$ (see 5.7.4.20), shall be inferred as the most recent DL-time whose residue under the negotiated DLPDU format would give rise to the DL-time residue conveyed by the received DLPDU; and

NOTE — This inference will cause the current DL-time to be inferred when the SD-parameters field of the received DLPDU does not convey octets of DL-time.

iii) the DL-time of reception of the DLPDU shall be used as the time of writing the buffer, $V_B(TW)$ (see 5.7.4.19).

2) If (1) does not apply, so that the received DLPDU conveyed a not-previously-received segment of a multi-segment DLSDU for which at least one segment had been previously received, then

i) the value of the T subfield from the newly-received DLPDU shall be ANDed into the buffer's associated timeliness-status, $V_B(TS)$ (see 5.7.4.21);

ii) any DL-time conveyed in the SD-parameter of the DLPDU shall be ignored; and

iii) the DL-time of reception of the DLPDU shall be used as the time of writing the buffer, $V_B(TW)$ (see 5.7.4.19).

9.2.2.6 Validation and processing of user-data received in a CA, DT or ED DLPDU

In the following, MUD is defined to be the maximum number of user-data octets which may be conveyed in a single CA, DT or ED DLPDU of the DLCEP's priority, as specified in 5.2.3.

If a received CA, DT or ED DLPDU has a non-null user data field following its SD-parameters field, then the contained data is the ASN'th segment, of TNS segments, of the *K*'th DLSDU, where ASN and TNS are both zero-origin, and *K* has the value last given it in 9.2.2.5.1(a.i) or 9.2.2.5.1(b.2.ii.A) or (b.2.iii) or (b.4).

The receiving DLE shall check whether (ASN × MUD plus the length of the received user-data) is less than or equal to the permitted maximum DLSDU size, $P_C(NP.M\ldots M_R)$, negotiated for this (receiving) direction of DLC transmission. If ASN is less than TNS, then the DLE shall also check whether the length of the received user-data is equal to MUD. If either of these requirements is violated, then the DLE shall disconnect the DLCEP as specified in 9.2.1.8, with a reason of "provider-originated disconnection — wrong DLSDU size, permanent condition:"

a) The receiving DLE shall determine whether a received-data record has already been allocated to the *K*'th DLSDU, and if not, shall allocate a record which may contain at least min($P_C(NP.M\ldots M_R)$, TNS × MUD) octets of DLS-user-data and shall associate that record with the *K*'th received DLSDU;

b) The receiving DLE shall copy the user-data field from the received DLPDU to that received-data record, starting with octet (ASN × MUD +1) of the received-data record; and

c) If the set variable $V_{C,K}(MRS)$ is now empty, then the receiving DLE shall attempt to deliver the DLSDU as specified in 9.2.2.7.

NOTE — It is possible for an implementation of this DL-protocol to perform DLSDU reassembly at the time of DLPDU receipt. Such an implementation approach could minimize the DLE's processing of DLS-user-data.

9.2.2.7 Delivery of an entire DLSDU which has been completely received at a DLCEP

Let *K* be the sequence number associated with the completely-received DLSDU, as determined by 9.2.2.5(A) or 9.2.2.5.1(b) or (d) at the time that DLSDU reception was completed.

If K is greater than $V_C(L)$, and the current state $V_C(ST)$ of the DLCEP is DATA TRANSFER READY, then the DLCEP's receiving data-delivery features and the type of receive buffer or queue binding determine the DLSDU delivery policy:

a) If the DLC is a CLASSICAL DLC, and K is greater than $V_C(L) + 1$, then the DLSDU has been received out of order, and the DLE shall retain but not deliver the DLSDU at this time.

b) Otherwise,

1) if a receiving buffer is bound to the DLCEP, then the DLE shall do as specified in 9.2.2.7.1;

2) if an explicit receiving queue is bound to the DLCEP, then the DLE shall do as specified in 9.2.2.7.2; or

3) if no receiving queue is bound to the DLCEP, which is the OSI default situation, then the implementation shall consistently either

i) do as specified in 9.2.2.7.3; or

ii) treat this as a variant of case (b), using a receiving queue which the implementation has assigned for this purpose.

NOTE — This queue may be unique to this DLCEP, or may be shared with other similar-priority DLCEPs at this DLSAP.

9.2.2.7.1 Delivery to a receive buffer

a) If the receiving DLCEP has a receiver timeliness attribute other than NONE, then the DLE shall set the variable $V_B(TW)$ (see 5.7.4.19) associated with writing the buffer to the current DL-time.

NOTE 1 — This is the local time of receipt, and not the remote time-of-production.

b) The DLE shall deliver the complete DLSDU as the new contents of the buffer, and shall associate any timeliness information received in the conveying DLPDU(s).

c) The DLE shall report a DL-BUFFER-RECEIVED indication to the DLS-user. If the receiving DLCEP-class was ORDERED, and the received DLSDU is a duplicate of a prior-received DLSDU, then the duplicated-DLSDU attribute of the DL-BUFFER-RECEIVED indication shall specify TRUE; in all other cases it shall specify FALSE.

If this DLCEP has been specified as a synchronizing DLCEP during the establishment of one or more other local DLCEPs, then the DLE shall record the DL-time of network access, $V_C(TNA)$, for use in the timeliness computations of those referencing DLCEP(s).

d) Any ongoing accesses to the contents of the buffer which are incomplete at the time of DLSDU delivery shall not be affected by the DLSDU delivery.

NOTE 2 — This constraint ensures that each access to a buffer is logically atomic.

e) The DLE shall set $V_C(L)$ equal to K; and shall cancel all timers $T_{C,N}(RRS)$ which may exist, where N is less than or equal to K. All DLE resources devoted to reception and reassembly of DLSDUs with sequence numbers less than or equal to K should be released.

9.2.2.7.2 Delivery to a receive queue

The DLE shall attempt to append the complete DLSDU, together with identification of the receiving DLCEP, to the receiving queue.

If unsuccessful, the DLE shall inform local DL-management of this queue-full situation.

NOTE — This DL-management notification may take the form of incrementing a counter of discarded DLSDUs.

If successful,

a) The DLE shall report a DL-DATA indication to the DLS-user.

b) The DLE shall cancel the timer $T_{C,K}$(RRS) if it exists.

c) If the DLC is an UNORDERED or ORDERED DLC, then

1) the DLE shall set V_C(L) equal to *K*; and

2) the DLE shall cancel all timers $T_{C,N}$(RRS) which may exist, where *N* is less than *K*. All DLE resources devoted to reception and reassembly of DLSDUs with sequence numbers less than or equal to *K* should be released.

d) If the DLC is a DISORDERED DLC, and if *K* equals (V_C(L) +1), then

1) the DLE shall set V_C(L) equal to *K;*

2) if *K* is less than V_C(H), then the DLE shall increment *K*. If the set variable $V_{C,K}$(MRS) is empty, then the DLE shall set V_C(L) equal to *K* and shall repeat this step; and

3) if the DLC is a PEER DLC, then if the DLE's DL-address unscheduled-service queue, Q(US), does not already contain a reference to the DLCEP, then the DLE shall append to that Q(US) a reference to the DLCEP, to ensure that an acknowledgment of DLSDU receipt is sent from the receiving DLCEP.

e) If the DLC is a CLASSICAL DLC, then

1) the DLE shall set V_C(L) equal to *K;*

2) if *K* is less than V_C(H), then the DLE shall increment *K*. If the set variable $V_{C,K}$(MRS) is empty, then the DLE shall repeat the entire data delivery procedure (see 9.2.2.7.2(a), (b) and (e)) using the new value of *K*; and

3) if the DLC is a PEER DLC, then if the DLE's DL-address unscheduled-service queue, Q(US), does not already contain a reference to the DLCEP, then the DLE shall append to that Q(US) a reference to the DLCEP, to ensure that an acknowledgment of DLSDU receipt is sent from the receiving DLCEP.

9.2.2.7.3 OSI-default delivery

The DLE shall report a DL-DATA indication to the DLS-user, conveying the received DLSDU as a parameter; after which the DLE shall do as specified in 9.2.2.7.2(b) through (e).

9.2.2.8 Receipt of a DT DLPDU addressed to a DLCEP

In the following, MUD is defined to be the maximum number of user-data octets which may be conveyed in a single DT DLPDU of the DLCEP's priority, as specified in 5.2.3.

When the DLE receives a DT DLPDU addressed to a DLCEP of the DLE, the DLE shall perform the following series of actions:

a) If the DLCEP state, V_C(ST), is WAITING-FOR-RESET-completion and the DLE is waiting only for receipt of a DT DLPDU at the DLCEP, then the DLE shall change the DLCEP state to DATA-TRANSFER-READY.

b) The DLE shall validate that

1) the priority of the received DT DLPDU is as expected;

2) in a received DT DLPDU addressed to all subscribers of a PUBLISHER DLCEP, the length of the publisher's DL-address is greater than or equal to that expected;

3) in a received DT DLPDU addressed to a PUBLISHER DLCEP, the number of DL-addresses is as expected;

4) that in a received DT DLPDU addressed to a PEER DLCEP;

i) the length and number of the DL-address(es) is as expected (only LONG; or only SHORT; or either SHORT or VERY-SHORT at the sender's option); and

ii) when two addresses are expected, that the second DL-address of the DLPDU is the DLCEP-address of the remote peer of the DLCEP addressed by the DT DLPDU's first DL-address.

If this validation fails, then

— if the DLCEP is PEER DLCEP, the DLE shall disconnect the DLCEP from the DLC as specified in 9.2.1.8 with a reason of "provider-originated disconnection — wrong DLPDU format or parameters, permanent condition;"

— else the DLE shall discard the DT DLPDU.

c) If the DLCEP is a PEER or SUBSCRIBER DLCEP whose negotiated residual-activity attribute is TRUE, then the DLE shall restart the DLCEP's T_C(RAM) as specified in 9.2.2.14.

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

d) If the remaining number of octets in the DLPDU is less than the number of octets in the negotiated SD-parameters format for the applicable sender-to-receiver direction of transmission, then

— if the DLCEP is PEER or SUBSCRIBER DLCEP, then the DLE shall disconnect the DLCEP from the DLC as specified in 9.2.1.8 with a reason of "provider-originated disconnection — wrong DLPDU format or parameters, permanent condition;"

— else the DLE shall discard the DT DLPDU.

Otherwise the DLE shall parse and process the applicable-format SD-parameters from those remaining octets as specified in 9.2.2.5; and if the remaining number of octets in the DLPDU, after the SD-parameters, is greater than zero, then the DLE shall process that user-data as specified in 9.2.2.6 and possibly 9.2.2.7.

9.2.2.9 Receipt of a CA, CD or ED DLPDU

When the DLE receives a CA, CD or ED DLPDU addressed to a DLCEP of the DLE, the DLE shall perform the following series of actions:

a) If the DLCEP is a PEER DLCEP then the DLE shall validate that

1) the length and number of DL-addresses is as expected;

2) when present and the receiving DLCEP is a PEER DLCEP, that the second DL-address of the DLPDU is the DLCEP-address of the remote peer of the receiving DLCEP; and

3) the DLL priority of the DLCEP is equal to the priority specified in the received DLPDU.

If this validation fails, then the DLE shall disconnect the DLCEP from the DLC as specified in 9.2.1.8 with a reason of "provider-originated disconnection — wrong DLPDU format or parameters, permanent condition."

b) If the DLCEP is a PEER DLCEP whose negotiated residual-activity attribute is TRUE, then the DLE shall restart the DLCEP's T_C(RAM) with a duration based on the user-specified maximum confirm delay for DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY primitives. If the specified value was other than UNLIMITED, then the duration of this timer should be ; otherwise the duration should be . DL-management may override these preferred durations.

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

c) If the received DLPDU is a CD DLPDU, then

1) If the DLCEP is a PEER DLCEP, and if the remaining number of octets in the DLPDU is not zero, and is not equal to the number of octets in the negotiated SD-parameters format for the applicable sender-to-receiver direction of transmission, then the DLE shall disconnect the DLCEP from the DLC as specified in 9.2.1.8 with a reason of "provider-originated disconnection — wrong DLPDU format or parameters, permanent condition."

2) Otherwise, when (1) does not apply, then

i) if the remaining number of octets in the DLPDU is equal to the number of octets in the negotiated SD-parameters format for the applicable sender-to-receiver direction of transmission, then the DLE shall parse and process the applicable-format SD-parameters from those remaining octets as specified in 9.2.2.5; and

ii) when the receiving DLCEP's features are ORDERED, and the DLCEP is bound to a sending buffer, and the DLSDU in the buffer was too large to transmit completely in the immediate response to the received CD DLPDU, then the DLE shall

A) for the appropriate *K*, modify the variable $V_{C,K}$(SS), based on the length of the *K*'th DLSDU, to indicate that all segments of the DLSDU except segment zero (0) require transmission;

B) append to the DLE's unscheduled-service queue, Q(US), a reference to the DLCEP's Q_A(UR). The reference signals the need to transmit the contents of the sending buffer which is bound to the DLCEP.

d) If the received DLPDU is a CA or ED DLPDU, then

1) If the remaining number of octets in the DLPDU is equal to or greater than the number of octets in the negotiated SD-parameters format for the applicable sender-to-receiver direction of transmission; then the DLE shall:

i) parse and process the applicable-format SD-parameters from those remaining octets as specified in 9.2.2.5; and

ii) then process any remaining octets of the DLSDU as received DLS-user-data as specified in 9.2.2.6 and 9.2.2.7.

2) If (1) does not apply and the DLCEP is a PEER DLCEP, then the DLE shall disconnect the DLCEP from the DLC as specified in 9.2.1.8 with a reason of "provider-originated disconnection — wrong DLPDU format or parameters, permanent condition."

9.2.2.10 Starting, cancellation and expiration of the timer T_U(MCD) on a DL-DATA request

The timer T_U(MCD) shall be started when the DLS-user issues the corresponding DL-DATA request. It shall be canceled

— at a subscriber DLCEP, or at a CLASSICAL or DISORDERED peer DLCEP, or at an ORDERED peer or publisher DLCEP when V_C(R) does not exist, when the DLE issues the corresponding DL-DATA confirm; and

— at a CLASSICAL or DISORDERED publisher DLCEP, or at an ORDERED peer or publisher DLCEP when $V_C(R)$ exists, when the DLE finds it necessary to increment $V_C(R)$ and the resultant value of $V_C(R)$ equals the sequence number assigned to the corresponding DLSDU.

If the timer $T_U(MCD)$ expires on a DL-DATA request, then the DLE shall

a) increment $V_C(R)$;

NOTE — After incrementation, $V_C(R)$ should have a value equal to the sequence number of the DLSDU associated with the expired request.

b) remove the request from the sending DLCEP-address's user-request queue, $Q_A(UR)$, and terminate processing of the request;

c) maintain any appropriate DL-management statistics;

NOTE — Determination of the minimum necessary statistics is for future study.

d) if a DL-DATA confirm primitive for the request has not yet been issued, then

1) initiate a DL-DATA confirm reporting "provider-originated failure — request timeout;" and

2) if the choice of sending DLCEP data delivery features is DISORDERED or CLASSICAL, then initiate a reset of the DLC as specified in 9.2.2.19.

9.2.2.11 Starting, cancellation and expiration of the timer $T_{C,K}(SS)$

NOTE — This timer is used only by PEER DLCEPs whose sending data delivery features are DISORDERED or CLASSICAL.

The timer $T_{C,K}(SS)$ shall be started whenever a DLPDU containing all or part of $DLSDU_K$ is transmitted and $V_{C,K}(SS)$ is empty; it shall be canceled whenever $V_C(A)$ is greater than or equal to K or whenever $V_{C,K}(SS)$ becomes non-empty (see 9.2.2.5.2(b)).

The duration of this timer shall be based on the local user-specified maximum confirm delay for DL-DATA primitives. If the specified value was other than UNLIMITED, then the duration of this timer should be between 25% and 50% of ; otherwise the duration should be between 25% and 50% of . DL-management may override these preferred durations.

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

If the timer $T_{C,K}(SS)$ expires, then the DLE shall

a) modify the variable $V_{C,K}(SS)$, based on the length of the K'th DLSDU, to indicate that the last segment of the K'th DLSDU need retransmission;

b) append to the DLE's unscheduled-service queue, Q(US), a reference to the DLCEP's $Q_A(UR)$, to schedule a retransmission of the unacknowledged DLSDU; and

c) maintain any appropriate DL-management statistics.

NOTE — Determination of the minimum necessary statistics is for future study.

9.2.2.11.1 Use of the simplified timer $T_C(SS)$

When the permission of 5.7.4.9.1 is employed, the following rules apply:

The timer $T_C(SS)$ shall be started, but not restarted, whenever a DLPDU containing all or part of $DLSDU_K$ is transmitted and $V_{C,K}(SS)$ is empty. The timer shall be restarted whenever it is not running and $V_C(A)$ is less than $V_C(M)$; it shall be canceled whenever $V_C(A)$ equals $V_C(M)$ or whenever $V_{C,K}(SS)$ becomes non-empty due to receipt of a request for retransmission (see 9.2.2.5.2(b)).

The duration of this timer shall be based on the local user-specified maximum confirm delay for DL-DATA primitives If the specified value was other than UNLIMITED, then the duration of this timer should be between 25% and 50% of ; otherwise the duration should be between 25% and 50% of . DL-management may override these preferred durations.

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

If the timer $T_C(SS)$ expires, then the DLE shall

a) modify the variable $V_{C,K}(SS)$, based on the length of the unacknowledged $DLSDU_K$ with the lowest sequence number, to indicate that the last segment of that DLSDU need retransmission;

b) append to the DLE's unscheduled-service queue, Q(US), a reference to the DLCEP's $Q_A(UR)$, to schedule a retransmission of the unacknowledged DLSDU; and

c) maintain any appropriate DL-management statistics.

9.2.2.12 Starting, cancellation and expiration of the timer $T_{C,K}(RRS)$

NOTE 1 — This timer is used by DISORDERED or CLASSICAL subscriber DLCEPs, and optionally by ORDERED peer or subscriber DLCEPs.

The timer $T_{C,K}(RRS)$ shall be started whenever it is not running and a DLPDU requesting retransmission of one or more segments of the *K*'th DLSDU is transmitted. It shall be canceled whenever all segments of the *K*'th DLSDU are received.

The duration of this timer shall be based on the remote user-specified maximum confirm delay for DL-DATA primitives and conveyed in an EC DLPDU previously-received from the sending DLCEP. If the specified value was other than UNLIMITED, then the duration of this timer should be between 25% and 50% of ; otherwise the duration should be between 25% and 50% of . DL-management may override these preferred durations.

NOTE 2 — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

If the timer $T_{C,K}(RRS)$ expires, then the DLE shall

a) set $V_{C,K}(RRS)$ equal to the then-current value of $V_{C,K}(MRS)$, to indicate that the specified segments of the *K*'th DLSDU need retransmission; and

b) check for a reference to the DLCEP on the DLE's unscheduled-service queue, Q(US), and if not found then append a reference to the DLCEP to the DLE's unscheduled-service queue, Q(US), to schedule a retransmission of the missing DLSDU segments; and

c) maintain any appropriate DL-management statistics.

NOTE — Determination of the minimum necessary statistics is for future study.

9.2.2.13 Starting, cancellation and expiration of the timer $T_C(RAS)$

NOTE 1 — This timer is used only by PUBLISHER or sending PEER DLEs whose sending data delivery features are ORDERED, DISORDERED or CLASSICAL, and is only required when DLC establishment requested residual activity on the DLC in this (sender to receiver) direction of data transfer.

When applicable (see 5.7.4.16 for the conditions of the timer's use), the timer $T_C(RAS)$ shall be started

a) at a sending DISORDERED or CLASSICAL PEER DLCEP, whenever it is not running and when $V_C(A)$ equals ($V_C(N) - 1$); and

b) at a PUBLISHER DLCEP, or sending ORDERED PEER DLCEP, whenever it is not running and when $V_C(M)$ equals ($V_C(N) - 1$) and $V_{C,VC(M)}(SS)$ is empty.

The duration of this timer shall be based on the user-specified maximum confirm delay for DL-CONNECT request or response primitives. If the specified value was other than UNLIMITED, then the duration of this timer should be between 70% and 95% of ; otherwise the duration should be between 70% and 95% of . DL-management may override these preferred durations.

NOTE 2 — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

It shall be canceled whenever $V_C(A)$ is not equal to ($V_C(N) – 1$). If the timer $T_C(RAS)$ expires, then the DLE shall check for a reference to the DLCEP on the DLE’s unscheduled-service queue, Q(US), and if not found then append a reference to the DLCEP to the DLE’s unscheduled-service queue, Q(US), to schedule a transmission to the remote DLCEP(s).

NOTE 3 — The resulting transmission may consist of a DT DLPDU with a null user-data field, or a CA DLPDU.

9.2.2.14 Starting, cancellation and expiration of the timer T_C(RAM)

NOTE 1 — This timer is used only by SUBSCRIBER or receiving PEER DLEs whose receiving data delivery features are ORDERED, DISORDERED or CLASSICAL, and is only required when DLC establishment requested residual activity on the DLC in this (receiver-from-sender) direction of data transfer.

When applicable (see 5.7.4.17 for the conditions of the timer’s use), the timer $T_C(RAM)$ shall run continuously. It shall be restarted whenever any DLPDU is received on the DLCEP.

The duration of this timer shall be based on the remote user-specified maximum confirm delay for DL-CONNECT request or response primitives and conveyed in an EC DLPDU previously-received from the sending DLCEP. If the specified value was UNLIMITED, then the duration of this timer should be 60 s. Otherwise, the duration should be $V_C(NP)$.MCD_CRS. DL-management may override these preferred durations.

If the timer $T_C(RAM)$ expires, then the DLE shall initiate a reset of the DLCEP as specified in 9.2.2.19.

9.2.2.15 Receipt of a DL-RESET request primitive

When the DLE receives a DL-RESET request from a DLS-user for a DLCEP in the DATA-TRANSFER-READY state, then

a) If the DLCEP class is not PEER, the DLE shall issue a confirm primitive with a status of “failure — reason unspecified” for each outstanding DL-DATA request primitive on the local DLCEP, and shall increment $V_C(R)$ for each such confirmed primitive.

b) The DLE shall

1) set $V_C(M)$ equal to $V_C(N)$ -1; set $V_C(MS)$ and $V_C(HS)$ equal to zero;

2) release all DLE resources devoted to reception and reassembly of DLSDUs; and

3) cancel all timers $T_{C,K}(SS)$ and $T_{C,K}(RRS)$.

c) If the DLCEP-class is PEER, then the DLE shall

1) note that both

i) a DL-RESET confirm primitive; and

ii) receipt of an RC DLPDU not requesting a reply RC DLPDU from the remote peer DLCEP

will be required;

2) encode an RC DLPDU, addressed to the remote peer DLCEP and requesting a reply RC DLPDU, with the reason specified by the DLS-user, using $V_C(L)$ +1for the NDR sequence number and $V_C(M)$for the NDS sequence number;

3) schedule the DLPDU for transmission at the DLC's priority as specified in 9.4.5;

4) start a timer as specified in 9.2.1.2(b); and

5) change the DLCEP state, $V_C(ST)$, to WAITING-FOR-RESET-COMPLETION, while waiting for an RC DLPDU.

d) If the DLCEP-class is PUBLISHER, then the DLE shall

1) encode an RC DLPDU, addressed to all of the DLC's subscriber's DLCEPs, not requesting a reply RC DLPDU, with the reason specified by the DLS-user, using zero (0) for the NDR sequence number and $V_C(M)$ for the NDS sequence number,

2) set $V_C(A)$ or $V_C(R)$, as appropriate, equal to $V_C(N)$ –1;

3) schedule the DLPDU for transmission at the DLC's priority as specified in 9.4.5; and

4) report a DL-RESET confirm to the local DLS-user with a status parameter of "success."

e) If the DLCEP-class is SUBSCRIBER, then the DLE shall report a DL-RESET confirm to the local DLS-user with a status parameter of "success."

9.2.2.16 Receipt of a DL-RESET response primitive

When the DLE receives a DL-RESET response from a DLS-user for an active DLCEP whose state is WAITING-FOR-RESET-COMPLETION then

a) If the DLCEP-class is PEER, and the set of events at the DLCEP for which the DLE is waiting is

1) only a DL-Reset response, then the DLE shall

i) encode a DT DLPDU (with or without data) from the DLCEP to notify its peer of the successful receipt of the confirming RC DLPDU;

ii) schedule the DLPDU for transmission at the DLC's priority as specified in 9.4.5;

iii) again flush the DLCEP's internal queues;

iv) change the DLCEP state, $V_C(ST)$, to DATA-TRANSFER-READY; and

v) report a DL-RESET-COMPLETED indication to the local DLS-user;

2) both

i) a DL-Reset response; and

ii) receipt of an RC DLPDU not requesting a reply from the peer DLCEP,

then the DLE shall wait for that RC DLPDU;

3) both

i) a DL-Reset response; and

ii) subsequent receipt of a DT DLPDU from the peer DLCEP,

then the DLE shall

A) encode an RC DLPDU, addressed to the remote peer DLCEP and not requesting a reply RC DLPDU, with a reason of “reset, unknown origin — reason unspecified,” using $V_C(L)$ +1 for the NDR sequence number and $V_C(M)$ for the NDS sequence number;

B) schedule the DLPDU for transmission at the DLC’s priority as specified in 9.4.5;

C) start a timer as specified in 9.2.1.2(b);

D) note that a DL-RESET-COMPLETED indication to the local DLS-user is still required; and

E) wait for a DT DLPDU.

b) If the DLCEP-class is PUBLISHER or SUBSCRIBER, then the DLE shall

1) change the DLCEP state, $V_C(ST)$, to DATA-TRANSFER-READY;

2) again flush the DLCEP’s internal queues; and

3) report a DL-RESET-COMPLETED indication to the local DLS-user.

9.2.2.17 Receipt of an RC DLPDU

When the DLE receives an RC DLPDU, the DLE shall determine the version of the DL-protocol in use, as specified in the received RC DLPDU, and shall interpret the other RC-parameters of the DLPDU accordingly:

a) If the DLCEP’s DLCEP-class is PEER or SUBSCRIBER, when the DLE receives an RC DLPDU specifying that the DLCEP should be reset, then the DLE shall take action depending on the DLCEP state, $V_C(ST)$:

1) WAITING-FOR-EC-DLPDU or WAITING-FOR-CONNECT-RESPONSE or WAITING-FOR-CONNECT-COMPLETION:

The DLE shall

i) disconnect the DLCEP as specified in 9.2.1.8 with a reason of “connection rejection — inconsistent DLCEP state, — permanent condition.”

2) DATA-TRANSFER-READY:

The DLE shall

i) report a DL-RESET indication to the local DLS-user with a reason parameter equal to that in the received RC DLPDU;

ii) If the DLCEP class is PEER and $V_C(A)$ is less than the sequence number implied by the NDR subfield of the RC parameters of the received RC DLPDU (which is the smallest sequence number $\geq V_C(A)$ whose modulus would be NDR); the DLE shall process outstanding DL–DATA requests on the local DLCEP as specified in 9.2.2.5.2, with NDR as received in the RC DLPDU, J=0 and K=1;

iii) if the DLCEP’s DLCEP-class is PEER, the DLE shall issue a confirm primitive with a status of “failure — reset or disconnection” for each outstanding DL-DATA request primitive on the local DLCEP, and shall increment $V_C(R)$ for each such confirmed primitive; and set $V_C(A)$ or $V_C(R)$, as appropriate, equal to $V_C(N)$ –1;

iv) set $V_C(L)$ and $V_C(H)$ equal to the sequence number implied by the NDS sub-field of the RC-parameters of the received RC DLPDU (which is the smallest sequence number $\geq V_C(H)$ whose modulus would be NDS);

v) set $V_C(M)$ equal to $V_C(N)$ -1; set $V_C(MS)$ and $V_C(HS)$ equal to zero;

vi) release all DLE resources devoted to reception and reassembly of DLSDUs;

vii) cancel all timers $T_{C,K}$(SS) and $T_{C,K}$(RRS);

viii) change the DLCEP state, V_C(ST), to WAITING-FOR-RESET-COMPLETION; and

ix) wait for a DL-RESET response plus, if the DLCEP class is PEER, subsequent receipt of a DT DLPDU indicating peer reset completion;

3) WAITING-FOR-RESET-COMPLETION:

i) If the DLCEP-class is PEER and V_C(A) is less than the sequence number implied by the NDR sub-field of the RC-parameters of the received RC DLPDU (which is the smallest sequence number $\geq V_C$(A) whose modulus would be NDR); the DLE shall process outstanding DL-DATA requests on the local DLCEP as specified in 9.2.2.5.2, with NDR as received in the RC DLPDU, J=0 and K=1.

ii) If the DLCEP's DLCEP-class is PEER, the DLE shall issue a confirm primitive with a status of "failure — reset or disconnection" for each outstanding DL-DATA request primitive on the local DLCEP, and shall increment V_C(R) for each such confirmed primitive; and set V_C(A) or V_C(R), as appropriate, equal to V_C(N) –1.

iii) The DLE shall set V_C(L) and V_C(H) equal to the sequence number implied by the NDS sub-field of the RC-parameters of the received RC DLPDU (which is the smallest sequence number $\geq V_C$(H) whose modulus would be NDS).

iv) If the DLCEP-class is SUBSCRIBER, then the DLE shall discard the received RC DLPDU.

v) If the DLCEP-class is PEER, and the DLE is waiting for a DL-RESET response primitive at the DLCEP, then
— if the received RC DLPDU specified that a reply RC DLPDU is required, then the DLE shall note that an RC DLPDU is required, else the DLE shall note that a DT DLPDU is required.

vi) If the DLCEP-class is PEER, and the DLE is not waiting for a DL-RESET response primitive at the DLCEP, and the received RC DLPDU requested a reply, then the DLE shall

A) encode an RC DLPDU, addressed to the remote peer DLCEP and not requesting a reply RC DLPDU, with a reason of "reset, unknown origin — reason unspecified," using V_C(L) +1 for the NDR sequence number and V_C(M) for the NDS sequence number;

B) schedule the DLPDU for transmission at the DLC's priority as specified in 9.4.5;

C) start a timer as specified in 9.2.1.2(b); and

D) if the reset-completion primitive expected by the DLS-user at the DLCEP is DL-RESET confirm, then the DLE shall note that an RC DLPDU is required, else the DLE shall note that a DT DLPDU is required.

vii) If the DLCEP-class is PEER, and the DLE is not waiting for a DL-RESET response primitive at the DLCEP, and the received RC DLPDU did not request a reply, then the DLE shall

A) encode a DT DLPDU (with or without data) from the DLCEP to notify its peer of the successful receipt of the confirming RC DLPDU;

B) schedule the DLPDU for transmission at the DLC's priority as specified in 9.4.5;

C) again flush the DLCEP's internal queues;

D) change the DLCEP state, V_C(ST), to DATA-TRANSFER-READY; and

E) if the reset-completion primitive expected by the DLS-user at the DLCEP is DL-RESET confirm, then the DLE shall report a DL-RESET confirm to the local DLS-user, else the DLE shall report a DL-RESET-COMPLETED indication to the local DLS-user.

viii)When none of (ii) through (v) applies, then

A) the DLE shall encode a DC DLPDU as specified in 7.2 and 8.2, with a reason of "provider-originated disconnection or connection rejection — reason unspecified,” and schedule the DC DLPDU for transmission at TIME-AVAILABLE priority as specified in 9.4.5; and

B) the DC DLPDU shall have both destination and source addresses (see 7.2.1 formats 1L and 1S), the destination address shall be identical to the first source DLCEP-address of the received RC DLPDU, and the source address shall be identical to the destination DL-address of that received RC DLPDU.

4) In all other cases the DLE shall act as in (a.3.v).

b) If the DLCEP's DLCEP-class is PUBLISHER, then the DLE shall act as in (a.3.v).

9.2.2.18 Expiration of the timer T_U(MCD) on a DL-RESET request or indication

If the timer T_U(MCD) expires on a DL-RESET request or DLE-initiated reset, then

a) if this is the (V(NRC)+1)'th consecutive expiration without returning to the DATA-TRANSFER-READY state, then the DLE shall terminate DLCEP-reset processing and shall disconnect the DLCEP from the DLC as specified in 9.2.1.8 with a reason of “provider-originated disconnection — timeout, permanent condition;”

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

b) otherwise, if (a) does not apply, then the DLE shall

1) restart the timer with the same period as the previous time; and

2) reschedule the RC DLPDU (which was first sent just after exiting the DATA-TRANSFER-READY state) for retransmission at the DLC's priority as specified in 9.4.5.

9.2.2.19 DLE-initiated reset

When the DLE initiates a reset, then

a) the DLE shall change the DLCEP state, V_C(ST), to WAITING-FOR-RESET-COMPLETION;

b) the DLE shall note the need for

1) receipt of a DL-RESET response from the DLS-user; and

2) if the DLCEP class is PEER, subsequent receipt of an RC DLPDU indicating peer reset completion;

c) if the DLCEP's DLCEP-class is PEER, then the DLE shall start a user-request timer T_U(MCD) with a duration based on the user-specified maximum confirm delay for DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY primitives. If the specified value was other than UNLIMITED, then the duration of this timer should be $\frac{60s}{V(NRC)+1}$; otherwise the duration should be $\frac{V_C(NP) \cdot MCD\_CRS}{V(NRC)+1}$. DL-management may override these preferred durations;

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

d) if the DLCEP's DLCEP-class is not PEER, the DLE shall issue a confirm primitive with a status of “failure — reset or disconnection” for each outstanding DL-DATA request primitive on the local DLCEP, shall increment V_C(R) for each such confirmed primitive, and shall set V_C(A) or V_C(R), as appropriate, equal to V_C(N) –1;

e) the DLE shall

1) set V_C(M) equal to V_C(N) -1; set V_C(MS) and V_C(HS) equal to zero;

2) release all DLE resources devoted to reception and reassembly of DLSDUs; and

3) cancel all timers $T_{C,K}$(SS) and $T_{C,K}$(RRS);

f) the DLE shall issue a DL-RESET indication primitive to the local DLS-user, specifying the reason for the reset; and

g) if the DLCEP class is PEER, then the DLE shall

1) encode an RC DLPDU, with the same reason as the reason given to the local DLS- user and requesting a reply RC DLPDU, using V_C(L) +1 for the NDR sequence number and V_C(M) for the NDS sequence number; and

2) queue that RC DLPDU for transmission at the DLC's priority as specified in 9.4.5; and

h) if the DLCEP-class is PUBLISHER, then the DLE shall

1) encode an RC DLPDU, addressed to all of the DLC's subscriber's DLCEPs, not requesting a reply RC DLPDU, with the same reason as the reason given to the local DLS- user, using zero (0) for the NDR sequence number and V_C(M) for the NDS sequence number, and

2) schedule the DLPDU for transmission at the DLC's priority as specified in 9.4.5.

9.2.3 Operation of the DLC subscriber query service

The primitives of the DLC subscriber query service are DL-SUBSCRIBER-QUERY request and confirm.

9.2.3.1 Receipt of a DL-SUBSCRIBER-QUERY request primitive

If the request is accepted, as indicated by a returned status of "request accepted" for the DL-SUBSCRIBER-QUERY request, then upon completion of the request, either successfully or after failure, the DLE shall issue a DL-SUBSCRIBER-QUERY confirm at the DLCEP, conveying the status of the request to the DLS-user.

The DL-SUBSCRIBER-QUERY request shall be queued on the DLE's NODE DL-address user-request queue, Q_N(UR), which is an implicit (DLE-controlled) queue:

a) If that queue is full, then the DLE shall immediately return the corresponding DL-LISTENER-QUERY confirm indicating the reason for failure — "provider-originated failure — queue full."

b) If (a) does not apply, then the DLE shall start a user-request timer T_U(MCD) with a duration based on the user-specified maximum confirm delay for DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY primitives. If the specified value was other than UNLIMITED, then the duration of this timer should be $\frac{V_c(NP) \cdot MCD_{-}CRS}{V(NRC)+1}$; otherwise, the duration should be $\frac{60s}{V(NRC)+1}$ .DL-management may override these preferred durations.

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

c) If (a) does not apply, then the DLE shall

— append the request to the DLE's NODE user-request queue, Q_N(UR), at TIME-AVAILABLE priority, where the request shall be placed in the second partition; and

NOTE — The third partition in the DLE's Q_N(UR) is always empty.

— append a reference to Q_N(UR) to the DLE's unscheduled-service queue, Q(US).

9.2.3.2 Transmission of a DL-address query SPDU

Upon receipt of a transmission opportunity for the queued DL-SUBSCRIBER-QUERY request, the DLE shall

a) form a DL-address query SPDU (see Annex B/B.3.6.2) specifying the requested DL-address; and

b) include that SPDU as the DLSDU of a connectionless format-1S DT DLPDU of TIME-AVAILABLE priority with a SHORT destination DL-address for the DL-support functions of all DLE's on the local link, 0100_{16} (see Annex A/A.3.2), and source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions.

9.2.3.3 Receipt of a DL-address query SPDU specifying a subscriber DLCEP-address

If a DLE receives a DL-address query SPDU/DLPDU for a DL-address and DL-address-class which is active within the DLE, then the DLE shall respond at a TIME-AVAILABLE priority with a corresponding DL-address reply SPDU/DLPDU, confirming the received DL-address query.

9.2.3.4 Receipt of a DL-address reply SPDU

If the DLE receives a DL-address reply SPDU/DLPDU in response to the DL-address query SPDU/DLPDU which was transmitted as a result of the DL-SUBSCRIBER-QUERY request, then the DLE shall

a) cancel the timer T_U(MCD);

b) initiate a DL-SUBSCRIBER-QUERY confirm reporting "success — a subscriber exists;" and

c) release the request identifier, making it available for subsequent reuse.

9.2.3.5 Expiration of the timer T_U(MCD) on a DL-SUBSCRIBER-QUERY request

If the timer T_U(MCD) expires on a DL-SUBSCRIBER-QUERY request,

a) if this is the (V(NRC)+1)'th consecutive expiration, then the DLE shall

   NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

   1) terminate processing of the request; and

   2) initiate a DL-SUBSCRIBER-QUERY confirm reporting "provider-originated failure — request timeout;"

b) otherwise, if (a) does not apply, then the DLE shall

   1) restart the timer with the same period as the previous time; and

   2) requeue the SPDU/DLPDU for retransmission as in 9.2.3.2(b).

9.3 Operation of the connectionless-mode services

The connectionless-mode services are connectionless data transfer with local-DLE-confirmation, connectionless data transfer with remote-DLE-confirmation, connectionless data exchange, and listener query.

9.3.1 Operation of the connectionless data transfer with local-DLE-confirmation service

The primitives of the connectionless data transfer with local-DLE-confirmation service are DL-UNITDATA request, indication and confirm.

9.3.1.1 Receipt of a DL-UNITDATA request primitive not specifying remote-DLE-confirmation

When the DLE receives a DL-UNITDATA request primitive not specifying remote-DLE-confirmation, it shall associate the DLS-user-specified request identifier with the request. If the request is accepted, as indicated by a returned status of "request accepted" for the DL-UNITDATA request, then upon completion of the requested transmission, either successfully or after failure, the DLE shall issue a DL-UNITDATA confirm with the same request identifier, conveying the status of the request to the DLS-user:

a) If the DL(SAP)-role of that source DL(SAP)-address is BASIC, then the calling-DL(SAP)-address specified in the DL-UNITDATA request is bound as a source at the specified priority to either an explicit (user-controlled) or implicit (DLE-controlled) queue.

b) If the DL(SAP)-role of that source DL(SAP)-address is not BASIC, or if that queue is full, or if the specified DLSDU has an invalid length, then the DLE shall immediately return the reason for failure as the status of the DL-UNITDATA request primitive.

c) If the called-DL(SAP)-address specified in the DL-UNITDATA request is not a DL(SAP)-address whose DL(SAP)-role is BASIC or GROUP, then the request is erroneous. The DLE is permitted, but is not required, to detect such an error (for example, by detecting that the called address is not a DL(SAP)-address). If such an error is detected, then the DLE shall immediately return the reason for failure as the status of the DL-UNITDATA request primitive.

d) Otherwise, if neither (b) nor (c) apply, then

   1) The DLE shall create and start a user-request timer T_U(MCD) with a duration based on the user-specified maximum confirm delay for the DL-UNITDATA request primitive. If the specified value was other than UNLIMITED, then the duration of this timer should be equal to that user-specified maximum confirm delay, P_U(MCD); otherwise the duration should be 60 s. DL-management may override these preferred durations.

   2) The DLE shall append the request to the calling DLSAP-address's user-request queue, Q_A(UR), as follows:

      i) If the DL-scheduling-policy attribute of the calling DLSAP-address is EXPLICIT, then the request shall await a DL-COMPEL-SERVICE request to release it for transmission and so shall be placed in the third partition of the user-request queue, Q_A(UR).

      ii) Otherwise, if (i) does not apply, then the request shall be placed in the second partition and the DLE shall append to the DLE's unscheduled-service queue, Q(US), a reference to Q_A(UR) of the same priority as the just-appended request.

9.3.1.2 Transmission of a unitdata DT DLSDU

Upon receipt of a transmission opportunity for the queued DLSDU, the DLE shall

a) remove the request from the appropriate DLSAP-address user-request queue, Q_A(UR);

b) form and send a DT DLPDU of the specified priority, with the specified called and calling DL(SAP)-addresses, with a null SD-parameters field (format P (see 8.4.1(a))), and with a user-data field whose length and contents equal the specified DLSDU;

   NOTE — With the exception of the final-token-use subfield of the DT DLPDU's first octet, the DT DLPDU may be formed when the request is queued, and need not be formed dynamically at the moment of transmission.

c) issue a DL-UNITDATA confirm with the associated request identifier reporting "success;" and

d) cancel and delete the associated timer T_U(MCD).

9.3.1.3 Receipt of a DT DLPDU, with an explicit source address, addressed to a DL(SAP)-address

When the DLE receives a DT DLPDU with an explicit source address addressed to a DL(SAP)-address bound to one or more of the DLE's DLSAPs, then if the DL(SAP)-role for that destination DL(SAP)-address is

a) INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER, then the received DLPDU is erroneous and the DLE

1) shall inform local DL-management of the event, possibly including the erroneous DLPDU's addresses; and

2) shall discard the DLPDU.

b) BASIC or GROUP, then for each of those bindings

1) the DLE shall attempt to append the received user-data as a DLSDU, together with the called and calling DL(SAP)-addresses and DLL priority of the received DLPDU, to

— the receiving queue which was bound at the received priority level; or

— to the implicit OSI queue if no receiving buffer or queue was bound at that received priority level;

2) if (1) is successful, then the DLE shall initiate a DL-UNITDATA indication primitive at that DLSAP;

3) otherwise, if (1) is not successful, then the DLE shall inform local DL-management of the queue-full situation.

NOTE — This DL-management notification may take the form of incrementing a counter of discarded DLSDUs.

9.3.1.4 Expiration of the timer T_U(MCD) on a DL-UNITDATA request not specifying remote-DLE-confirmation

If the timer T_U(MCD) expires on a DL-UNITDATA request not specifying remote-DLE-confirmation, then the DLE shall

a) remove the request from the appropriate DLSAP-address user-request queue, Q_A(UR), and the reference to the request from the DLE's unscheduled-service queue, Q(US); and

b) initiate a DL-UNITDATA confirm with the associated request identifier reporting "failure — timeout before transmission."

9.3.2 Operation of the connectionless data transfer service with remote-DLE-confirmation

The primitives of the connectionless data transfer with remote-DLE-confirmation service are DL-UNITDATA request, indication and confirm.

9.3.2.1 Receipt of a DL-UNITDATA request primitive specifying remote-DLE-confirmation

When the DLE receives a DL-UNITDATA request primitive specifying remote-DLE-confirmation, it shall associate the DLS-user-specified request identifier with the request. Upon completion of the request, either successfully or after failure, the DLE shall issue a DL-UNITDATA confirm with the same request identifier, conveying the status of the request to the DLS-user.

The calling DL(SAP)-address specified in the DL-UNITDATA request is bound as a source at the specified priority to either an explicit (user-controlled) or implicit (DLE-controlled) queue. If the DL(SAP)-role of that source DL(SAP)-address is not BASIC, or if that queue is full, or if the specified DLSDU has an invalid length, then the DLE shall immediately return the reason for failure as the status of the DL-UNITDATA request primitive.

If the called-DL(SAP)-address specified in the DL-UNITDATA request is not a DLSAP-address whose DL(SAP)-role is BASIC, then the request is erroneous. The DLE is permitted, but not required, to detect such an error (for example, by detecting that the called address is not a DLSAP-address). If such an error is detected during request processing, then the DLE shall return an appropriate error status in the DL-UNITDATA request primitive, indicating the reason for failure, and shall terminate processing of the request.

Otherwise

a) The DLE shall create and start a user-request timer T_U(MCD) with a duration based on the user-specified maximum confirm delay for the DL-UNITDATA request primitive. If the specified value was other than UNLIMITED, then the duration of this timer should be equal to that user-specified maximum confirm delay, P_U(MCD); otherwise the duration should be 60 s. DL-management may override these preferred durations.

b) The DLE shall append the request to the calling DLSAP-address' user-request queue, Q_A(UR), as follows:

   1) If the DL-scheduling-policy attribute of the calling DLSAP-address is EXPLICIT, then the request shall await a DL-COMPEL-SERVICE request to release it for transmission and so shall be placed in the third partition of the user-request queue, Q_A(UR).

   2) Otherwise the request shall be placed in the second partition of the user-request queue, Q_A(UR), and the DLE shall append to the DLE's unscheduled-service queue, Q(US), a reference to Q_A(UR) of the same priority as the just-appended request.

9.3.2.2 Transmission of a unitdata CA DLPDU

Upon receipt of a transmission opportunity for the queued DLSDU, the DLE shall form and send a CA DLPDU of the specified priority,

a) with the specified called and calling DLSAP-addresses;

b) with an SD-parameters field of format R (see 8.4.1(b)); and

c) with a user-data field whose length and contents equal the specified DLSDU.

NOTE — With the exception of the final-token-use subfield of the CA DLPDU's first octet, and possibly of the initiator's transaction index subfield, N(LTI), of the CA DLPDU's SD-parameters (see 8.4.1(b.3)), the CA DLPDU may be formed when the request is queued, and need not be formed dynamically at the moment of transmission.

9.3.2.3 Receipt of a CA DLPDU, with an explicit source address, addressed to a DL(SAP)-address

When the DLE receives a CA DLPDU with an explicit source address addressed to a DL(SAP)-address bound to one or more of the DLE's DLSAPs, then if the DL(SAP)-role for that destination DL(SAP)-address is

a) GROUP, then the received DLPDU is erroneous, and

   1) the DLE shall inform local DL-management of the event, possibly including the erroneous DLPDU's addresses; and

   2) the DLE shall discard the received CA DLPDU.

b) INITIATOR or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER, then the received DLPDU is erroneous and

1) the DLE shall inform local DL-management of the event, possibly including the erroneous DLPDU's addresses;

2) the DLE shall form and send, as an immediate reply, a reply DT DLPDU, as specified in 7.4.4.1 and 7.7, whose reported status indicates "failure —responder DL(SAP)-role incompatible with this DLPDU;" and

3) the DLE shall discard the received CA DLPDU.

c) BASIC, then

1) the DLE shall attempt to append the received user-data as a DLSDU, together with the called and calling DLSAP-addresses and DLL priority of the received DLPDU, to the receiving queue which was explicitly or implicitly bound at the received priority level;

2) if (1) is successful, then the DLE shall initiate a DL-UNITDATA indication primitive at that DLSAP and shall indicate its success in (4);

3) otherwise, if (1) is not successful, then the DLE shall indicate the reason for its failure in (4), and shall notify local DL-management of that failure;

NOTE — This DL-management notification may take the form of incrementing a counter of discarded DLSDUs.

4) the DLE shall form and send, as an immediate reply, a reply DT DLPDU, as specified in 7.4.4.1 and 7.7, whose reported status indicates the success (2) or failure of (3).

9.3.2.4 Receipt of a DT DLPDU, with an implicit source address, addressed to a DL(SAP)-address

When the DLE receives a DT DLPDU, with an implicit source address, addressed to a DL(SAP)-address bound to one or more of the DLE's DLSAPs, then if the DL(SAP)-role for that destination DL(SAP)-address is

a) CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER or GROUP, then the received DLPDU is erroneous; and

1) the DLE shall inform local DL-management of the event, possibly including the erroneous DLPDU's addresses; and

2) the DLE shall discard the received DT DLPDU.

b) INITIATOR, then the DLE shall act as specified in 9.3.3.3.

c) BASIC, then

1) If the transaction index subfield, N(LTI), of the SD-parameters of the received DLPDU specifies a transaction index for an outstanding incomplete request, then the DLE

i) shall release that transaction index, making it available for subsequent reuse;

ii) shall remove the request from the appropriate DLSAP-address user-request queue, Q_A(UR);

iii) shall cancel and delete the timer T_U(MCD) associated with the request; and

iv) shall issue a DL-UNITDATA confirm to the requesting DLS-user, indicating the request's DLS-user-identifier and reporting the status conveyed by the received DT DLPDU.

2) Otherwise, if (1) does not apply, then the DLE shall notify local DL-management of the receipt of an inappropriate DLPDU.

NOTE — This DL-management notification may take the form of incrementing a counter of discarded DLSDUs.

9.3.2.5 Receipt of an SR DLPDU

When the DLE receives an SR DLPDU,

— which reports a status other than “BOK — bridge OK” (that is, has a value other than E_{16}); and

— which is received as a reply to an immediately prior CA, CD or ED DLPDU which was originated by the receiving DLE and which was addressed to a DLSAP-address; and

— where the DL(SAP)-role for the DLSAP-address from which the transaction initiated (that is, the source DLSAP-address of the initiating CA, CD or ED DLPDU) is BASIC,

then the receiving DLE shall

a) release the transaction index, N(LTI), used in the immediately prior CA, CD or ED DLPDU, making it available for subsequent reuse;

b) remove the request from the appropriate DLSAP-address user-request queue, Q_A(UR);

c) cancel and delete the timer T_U(MCD) associated with the request; and

d) issue a DL-UNITDATA confirm to the requesting DLS-user, indicating the request’s DLS-user-identifier and reporting the status conveyed by the received SR DLPDU.

9.3.2.6 Expiration of the timer T_U(MCD) on a DL-UNITDATA request specifying remote-DLE-confirmation

If the timer T_U(MCD) expires on a DL-UNITDATA request specifying remote-DLE-confirmation, then the DLE

a) shall release the corresponding transaction index and remove it from active use, making it available for subsequent reuse only after a period of twice the maximum network DLPDU lifetime, V(NDL), of the extended link;

b) shall remove the request from the appropriate DLSAP-address user-request queue, Q_A(UR);

c) shall delete the timer T_U(MCD); and

d) shall initiate a DL-UNITDATA confirm to the requesting DLS-user, indicating the request’s DLS-user-identifier and reporting a status of “failure — timeout before transmission.”

9.3.3 Operation of the connectionless data exchange service

9.3.3.1 Transmission of a unitdata CD or ED DLSDU

Upon receipt of a transmission opportunity for a reference to a DL-address user-request queue, Q_A(UR), for a DLSAP-address whose DL(SAP)-role is INITIATOR, the DLE shall examine the set of sending buffer bindings for that DLSAP-address in priority order as follows:

a) If

— there is a sending binding at URGENT priority to a buffer; and

— the buffer is non-empty,

then the DLS-user-data contained in the URGENT buffer becomes the DLSDU selected for transmission, and the priority of that DLSDU is URGENT; or

b) If (a) does not apply, and

— the transaction-priority specified in the invoking reference is NORMAL or TIME-AVAILABLE; and

— there is a sending binding at NORMAL priority to a buffer; and

— the buffer is non-empty,

then the DLS-user-data contained in the NORMAL buffer becomes the DLSDU selected for transmission, and the priority of that DLSDU is NORMAL; or

c) If neither (a) nor (b) apply, and

— the transaction-priority specified in the invoking reference is TIME-AVAILABLE; and

— there is a sending binding at TIME-AVAILABLE priority to a buffer; and

— the buffer is non-empty,

then the DLS-user-data contained in the TIME-AVAILABLE buffer becomes the DLSDU selected for transmission, and the priority of that DLSDU is TIME-AVAILABLE; or

d) If none of (a) through (c) applies, then no DLSDU is selected for transmission, then the DLE shall form and send an appropriate DLPDU:

e) If any of (a) through (c) apply, then a DLSDU was selected by this procedure, and

1) the DLE shall form and send an ED DLPDU as specified in 7.6.3

i) with the destination address equal to the remote DLSAP-address specified in the invoking reference;

ii) with the source address equal to the DLSAP-address associated with the user-request queue, Q_A(UR), specified by that reference;

iii) with a priority equal to the transaction-priority specified in the invoking reference;

iv) with an SD-parameters field of format R (see 8.4.1(b)), specifying the actual priority of the selected DLSDU; and

v) with a user-data field whose length and contents equal the selected DLSDU; and

2) the DLE shall await an immediate response. If

i) the DLSDU which was sent to the responder in the ED DLPDU was obtained from a non-retentive buffer (BUFFER-NR); and

ii) an appropriate DT or SR DLPDU is received as an immediate reply, as specified in 7.5.3, 7.5.4, 7.6.3, 7.6.4, 7.7.3 and 7.8.3; and

iii) the explicit or implicit status conveyed by that reply is either “success” or “interim success,”

then the buffer shall be set empty.

f) If (d) applies, then no DLSDU was selected by this procedure, and

1) the DLE shall form and send a CD DLPDU as specified in 7.5.3

i) with the destination address equal to the remote DLSAP-address specified in the invoking reference;

ii) with the source address equal to the DLSAP-address associated with the user-request queue, Q_A(UR), specified by that reference;

iii) with a priority equal to the transaction-priority specified in the invoking reference; and

iv) with an SD-parameters field of format R (see 8.4.1(b)), specifying that there is no accompanying DLSDU; and

2) the DLE shall await an immediate response.

9.3.3.2 Receipt of a CD or ED DLPDU, with an explicit source address, addressed to a DL(SAP)-address

When the DLE receives a CD or ED DLPDU with an explicit source address addressed to a DL(SAP)-address bound to one or more of the DLE's DLSAPs, then if the DL(SAP)-role for that destination DL(SAP)-address is

a) GROUP, then the received DLPDU is erroneous, and

1) the DLE shall inform local DL-management of the event, possibly including the erroneous DLPDU's addresses; and

2) the DLE shall discard the received CD or ED DLPDU.

b) BASIC or INITIATOR, then the received DLPDU is erroneous, and

1) the DLE shall inform local DL-management of the event, possibly including the erroneous DLPDU's addresses;

2) the DLE shall form and send, as an immediate reply, a reply DT DLPDU, as specified in 7.5.4.1, 7.6.4.1 and 7.7, whose reported status indicates "failure —responder DL(SAP)-role incompatible with this DLPDU;"

3) the DLE shall discard the received CD or ED DLPDU.

c) CONSTRAINED RESPONDER, then

1) if the source address of the received CD or ED DLPDU equals the value of the remote-DLSAP-address parameter which was specified in the most recent DL-BIND request primitive for the responding DLSAP-address, then the DLE shall act as in (d);

2) otherwise, when (1) does not apply, the received DLPDU is erroneous and the DLE shall act as in (b) except that the status returned in (b.2) shall be "failure — response restricted to a different peer DLSAP-address."

d) UNCONSTRAINED RESPONDER, then

1) If the received DLPDU is an ED DLPDU and thus conveys a non-null DLSDU, then

i) If

— the responding DLSAP-address has an explicit binding as receiver at the priority of the received DLPDU; and

— the binding is to a buffer; and

— the size of the buffer is at least as great as the size of the DLSDU,

then the DLE shall overwrite the buffer with the just-received DLSDU.

ii) If

— the responding DLSAP-address has an explicit binding as receiver at the priority of the received DLPDU; and

— the binding is to a queue; and

— the size of each queue record is at least as great as the size of the DLSDU; and

— the queue is not full,

then the DLE shall append the just-received DLSDU to the queue.

iii) if either (i) or (ii) applies, then the DLE shall examine the set of sending buffer bindings for that DLSAP-address in priority order as follows:

A) if

— there is a sending binding at URGENT priority to a buffer; and

— the buffer is non-empty,

then the DLS-user-data contained in the URGENT buffer becomes the DLSDU selected for transmission, and the priority of that DLSDU is URGENT; or

B) if (A) does not apply, and

— the transaction-priority specified in the received DLPDU is NORMAL or TIME-AVAILABLE; and

— there is a sending binding at NORMAL priority to a buffer; and

— the buffer is non-empty,

then the DLS-user-data contained in the NORMAL buffer becomes the DLSDU selected for transmission, and the priority of that DLSDU is NORMAL; or

C) if neither (A) nor (B) apply, and

— the transaction-priority specified in the received DLPDU is TIME-AVAILABLE; and

— there is a sending binding at TIME-AVAILABLE priority to a buffer; and

— the buffer is non-empty,

then the DLS-user-data contained in the TIME-AVAILABLE buffer becomes the DLSDU selected for transmission, and the priority of that DLSDU is TIME-AVAILABLE; or

D) if none of (A) through (C) applies, then no DLSDU is selected for transmission.

iv) If either (i) or (ii) applies, then (iii) applied, and

A) the DLE shall form and send as an immediate reply an appropriate DT DLPDU as specified in 7.7;

B) if a DLSDU was sent to the initiator in a replying DT DLPDU, and that DLSDU was obtained from a non-retentive buffer (BUFFER-NR), then the buffer shall be set empty; and

C) the DLE shall issue a DL-UNITDATA-EXCHANGE indication to the responding DLS-user, reporting

— the responding DLSAP-address's DLS-user-/DL-identifier;

— the priority of the DLS-user-data conveyed by the received ED DLPDU, or that no such DLS-user-data was received (in a CD DLPDU);

— the priority of the DLS-user-data conveyed in the reply DT DLPDU, or that no such DLS-user-data was sent; and

— a status of "success."

v) if neither (i) nor (ii) applies, then the received DLSDU was discarded, and

A) the DLE shall form and send as an immediate reply an appropriate DT DLPDU as specified in 7.7; and

B) the DLE shall issue a DL-UNITDATA-EXCHANGE indication to the responding DLS-user, reporting

— the responding DLSAP-address's DLS-user-/DL-identifier;

— the priority of the DLS-user-data conveyed by the received ED DLPDU which was discarded; and

— a status of “failure.”

2) if the received DLPDU is a CD DLPDU and thus did not convey a DLSDU, then

i) the DLE shall examine the set of sending buffer bindings for that DLSAP-address in priority order as in (d.1.iii);

ii) the DLE shall act as in (d.1.iv.A) and (d.1.iv.B); and

iii) if

— a DLSDU was selected in (ii); or

— the indicate-null-UNITDATA-EXCHANGE-transactions parameter which was specified in the most recent DL-BIND request primitive for the responding DLSAP-address had the value TRUE,

then the DLE shall act as in (d.1.iv.C) except that the reported status shall be “failure — no DLS-user-data exchanged.”

9.3.3.3 Receipt of a DT DLPDU, with an implicit source address, addressed to a DL(SAP)-address

When the DLE receives a DT DLPDU, with an implicit source address, addressed to a DL(SAP)-address bound to one or more of the DLE’s DLSAPs, then if the DL(SAP)-role for that destination DL(SAP)-address is

a) BASIC or CONSTRAINED RESPONDER or UNCONSTRAINED RESPONDER or GROUP, then the DLE shall act as specified in 9.3.2.4.

b) INITIATOR, and if the transaction index subfield, N(LTI), of the SD-parameters of the received DLPDU specifies a transaction index for an outstanding incomplete request, then

1) The DLE shall release that transaction index, making it available for subsequent reuse.

2) The DLE shall cancel and delete the timer T_U(MCD) associated with the now-complete unit-data-exchange transaction.

3) If the invoking reference was in the DLE’s unscheduled-service queue, Q(US), or had been dynamically appended to a specified scheduled sequence, then the DLE shall remove that reference from that queue or sequence appendage.

4) If the received DT DLPDU contains DLS-user-data and thus conveys a non-null DLSDU, then

i) If

— the initiating DLSAP-address has an explicit binding as receiver at the priority of the received DLPDU; and

— the binding is to a buffer; and

— the size of the buffer is at least as great as the size of the DLSDU,

then the DLE shall overwrite the buffer with the just-received DLSDU.

ii) If

— the initiating DLSAP-address has an explicit binding as receiver at the priority of the received DLPDU; and

— the binding is to a queue; and

— the size of each queue record is at least as great as the size of the DLSDU; and

— the queue is not full,

then the DLE shall append the just-received DLSDU to the queue.

iii) If either (i) or (ii) applies, then

A) if a DLSDU which was sent to the responder in an ED DLPDU, and that DLSDU was obtained from a non-retentive buffer (BUFFER-NR), then the buffer shall be set empty.

B) the DLE shall issue a DL-UNITDATA-EXCHANGE indication to the initiating DLS-user, reporting

— the initiating DLSAP-address's DLS-user-/DL-identifier;

— the priority of the DLS-user-data conveyed by the previously-sent ED DLPDU, or that no such DLS-user-data was sent (in a CD DLPDU);

— the priority of the DLS-user-data conveyed by the received DLPDU; and

— a status of "success."

iv) If neither (i) nor (ii) applies, then the DLE shall issue a DL-UNITDATA-EXCHANGE indication to the initiating DLS-user, reporting

— the initiating DLSAP-address's DLS-user-/DL-identifier;

— the priority of the DLS-user-data conveyed by the previously-sent ED DLPDU, or that no such DLS-user-data was sent (in a CD DLPDU);

— the priority of the DLS-user-data conveyed by the received DLPDU; and

— a status of "failure — resource limitation in initiator."

5) If (4) does not apply, and the DLPDU sent to the responding DLE was an ED DLPDU, thus conveying a non-null DLSDU, then

i) If the DLSDU which was sent to the responder in the ED DLPDU was obtained from a non-retentive buffer (BUFFER-NR), then the buffer shall be set empty.

ii) The DLE shall issue a DL-UNITDATA-EXCHANGE indication to the initiating DLS-user, reporting

— the initiating DLSAP-address's DLS-user-/DL-identifier;

— the priority of the DLS-user-data conveyed by the previously-sent ED DLPDU, or that no such DLS-user-data was sent (in a CD DLPDU); and

— the status conveyed by the received DT DLPDU.

6) If neither (4) nor (5) apply, and the DLPDU sent to the responding DLE was a CD DLPDU, then no DLSDUs were exchanged. In this case, if the indicate-null-UNITDATA-EXCHANGE-transactions parameter which was specified in the most recent DL-BIND request primitive for the initiating DLSAP-address had the value TRUE, then the DLE shall issue a DL-UNITDATA-EXCHANGE indication to the initiating DLS-user, reporting the initiating DLSAP-address's DLS-user-/DL-identifier and a status of "failure — no DLS-user-data exchanged."

c) INITIATOR, and if the transaction index subfield, N(LTI), of the SD-parameters of the received DLPDU does not specify a transaction index for an outstanding incomplete request, then the DLE shall notify local DL-management of the receipt of an inappropriate DLPDU.

NOTE — This DL-management notification may take the form of incrementing a counter of discarded DLSDUs.

9.3.3.4 Receipt of an SR DLPDU

When the DLE receives an SR DLPDU,

— which reports a status other than “BOK — bridge OK” (that is, has a value other than E_{16});

— which was received as a reply to an immediately prior CA, CD or ED DLPDU which was originated by the receiving DLE and which was addressed to a DLSAP-address; and

— where the DL(SAP)-role for the DLSAP-address from which the transaction initiated (that is, the source DLSAP-address of the initiating CA, CD or ED DLPDU) is INITIATOR,

then

a) the receiving DLE shall release the transaction index, N(LTI), used in the immediately prior CA, CD or ED DLPDU, making it available for subsequent reuse;

b) the receiving DLE shall cancel and delete the timer T_U(MCD) associated with the request;

c) if the invoking reference was in the DLE’s unscheduled-service queue, Q(US), or had been dynamically appended to a specified scheduled sequence, then the DLE shall remove that reference from that queue or sequence appendage; and

d) if the indicate-null-UNITDATA-EXCHANGE-transactions parameter which was specified in the most recent DL-BIND request primitive for the initiating DLSAP-address had the value TRUE, then the DLE shall issue a DL-UNITDATA-EXCHANGE indication to the initiating DLS-user, reporting the initiating DLSAP-address’s DLS-user-/DL-identifier and a status of “failure — no DLS-user-data exchanged.”

9.3.3.5 Expiration of the timer T_U(MCD) on an incomplete instance of the unitdata exchange service

If the timer T_U(MCD) expires on an instance of the unitdata-exchange service, then the DLE

a) shall release the corresponding transaction index and remove it from active use, making it available for subsequent reuse only after a period of twice the maximum DLPDU lifetime, V(MDL), of the extended link;

b) shall delete the timer T_U(MCD); and

c) shall issue a DL-UNITDATA-EXCHANGE indication to the initiating DLS-user, specifying the initiating DLSAP-address’s DLS-user-/DL-identifier and reporting a status of “provider-originated failure — request timeout.”

9.3.4 Operation of the listener query service

The primitives of the listener query service are DL-LISTENER-QUERY request and confirm.

9.3.4.1 Receipt of a DL-LISTENER-QUERY request primitive

When the DLE receives a DL-LISTENER-QUERY request, it shall associate the DLS-user-specified request identifier with the request. If the request is accepted, as indicated by a returned status of “request accepted” for the DL-LISTENER-QUERY request, then upon completion of the request, either successfully or after failure, the DLE shall issue a DL-LISTENER-QUERY confirm with the same request identifier, conveying the status of the request to the DLS-user.

The DL-LISTENER-QUERY request shall be queued on the DLE’s NODE DL-address user-request queue, Q_N(UR), which is an implicit (DLE-controlled) queue:

a) If that queue is full, then the DLE shall immediately return the corresponding DL-LISTENER-QUERY confirm indicating the reason for failure — “provider-originated failure — queue full.”

b) If (a) does not apply, then the DLE shall start a user-request timer T_U(MCD) with a duration based on the user-specified maximum confirm delay for the DL-LISTENER-QUERY request primitive. If the specified value was other than UNLIMITED, then the duration of this timer should be ; otherwise the duration should be . DL-management may override these preferred durations.

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

c) If (a) does not apply, then the DLE shall

— append the request to the DLE's NODE user-request queue, Q_N(UR), at TIME-AVAILABLE priority, where the request shall be placed in the second partition; and

NOTE — The third partition in the DLE's Q_N(UR) is always empty.

— append a reference to Q_N(UR) to the DLE's unscheduled-service queue, Q(US).

9.3.4.2 Transmission of a DL-address query SPDU

Upon receipt of a transmission opportunity for the queued DL-LISTENER-QUERY request, the DLE shall form and send a DL-address query SPDU as specified in 9.2.3.2.

9.3.4.3 Receipt of a DL-address query SPDU specifying a group DL-address

If a DLE receives a DL-address query SPDU (conveyed by a DT DLPDU) for a DL-address and DL-address -class which is active within the DLE, then the DLE shall

a) form a replying DT DLPDU containing a corresponding DL-address reply SPDU, confirming the received DL-address query;

b) append the DT DLPDU to the DLE's NODE user-request queue, Q_N(UR), at the relevant priority; and

c) append a reference to Q_N(UR) to the DLE's unscheduled-service queue, Q(US).

9.3.4.4 Receipt of a DL-address reply SPDU

If the DLE receives a DL-address reply SPDU (conveyed by a DT DLPDU) as an apparent response to the DL-address query SPDU/DLPDU which was transmitted as a result of the DL-LISTENER-QUERY request, then the DLE shall

a) cancel the timer T_U(MCD); and

b) initiate a DL-LISTENER-QUERY confirm with the corresponding request identifier reporting “success — a listener exists.”

9.3.4.5 Expiration of the timer T_U(MCD) on a DL-LISTENER-QUERY request

If the timer T_U(MCD) expires on a DL-LISTENER-QUERY request, then

a) If this is the (V(NRC)+1)’th consecutive expiration, then the DLE shall

NOTE — See the permission at the start of clause 9 with regard to use of V(NRC) versus V(MRC).

1) terminate processing of the request; and

2) initiate a DL-LISTENER-QUERY confirm with the corresponding request identifier reporting “provider-originated failure — request timeout.”

b) Otherwise, if (a) does not apply, then the DLE shall

1) restart the timer with the same period as the previous time; and

2) requeue the same DT DLPDU for retransmission as specified in 9.3.4.2(b).

9.4 Operation of the scheduling guidance services

The scheduling guidance services are the DL-time service, the compel-service service, the sequence scheduling service, and the sequence subsetting service.

9.4.1 Operation of the DL-time service

The only primitive of the DL-time service is DL-TIME request.

9.4.1.1 Receipt of a DL-TIME request primitive

When the DLE receives a DL-TIME request, it shall respond with a five-component record as shown in Table 69, whose values are

a) the current value of the time source's link-id, V(TSL);

b) the quality of the current DL-time, V(TQ), as defined in 8.6(b);

c) the current value of the local DL-time-offset, V(DLTO);

d) the current value of the local-link-scheduling-time-offset, V(LSTO); and

e) the current value of the local node-time, C(NT).

The DLS-user may then sum the last three values to determine the node's current DL-time, or use the values separately, as appropriate.

Table 69 — Components of returned DL-time

octet	contents
1 2	V(TSL)
3	V(TQ)
4 5 6 7 8 9 10	V(DLTO)
11 12 13 14 15 16 17	V(LSTO)
18 19 20 21 22 23 24	C(NT)

If the DLE's time-synchronization class (see 11.3(a)) is NONE and the DLE does not maintain even an estimated C(NT), then it may respond with the values 0…0 for V(TQ), V(DLTO), V(LSTO) and C(NT), and $FFFF_{16}$ for V(TSL); this latter, which is an invalid link-id, indicates to the DLS-user the DLE's incapability.

9.4.1.2 Transmission of a TD DLPDU

After sending a TD DLPDU, the LAS DLE shall restart its local timer T(TDP), to expire at a time no greater than 0,95 × V(TDP) in the future (see 5.7.1.25 and 5.7.1.26).

9.4.1.3 Receipt of a TD DLPDU

If the DLE's time-synchronization class (see 11.3(a)) is NONE, then

a) If the DLE which is not the LAS DLE supports the variables, timers and counters defined in 5.7.1.19 through 5.7.1.27, then the DLE shall set V(TSL) equal to N_S(TSL), set V(DLTO) equal to N_S(DLTO), and set V(LSTO) equal to

$$(N_S(DLT) + N_S(DLTA) - N_S(DLTO) \; - N_R(NT) + \;) \text{ modulo } 2^{56} . \qquad \text{(Eq. 19)}$$

b) Otherwise, when (a) does not apply, then the DLE shall ignore receipt of the TD DLPDU.

When a DLE other than the LAS DLE, with a time-synchronization class other than NONE, receives a TD DLPDU, then

c) The DLE shall restart its local timer T(TDP), to expire at a time such that if it does not receive a TD DLPDU until after the expiration of this timer, it can still maintain time synchronization within the limits implied by the local link's time-synchronization-class (see 5.7.1.25). The value of this timer shall be larger than the minimum specified for the time-synchronization-class in 11.3(a), but the exact value may be determined locally by the DLE.

d) The DLE shall note that the DLE does not have to send a CT.

NOTE — Variance in the expiration periods of the timers T(TDP) among DLEs, and cancellation upon receipt of a TD DLPDU of such requests for transmission of a CT DLPDU, both serve to reduce the likelihood that a single link-wide noise event, inhibiting correct reception of a TD DLPDU, will result in multiple transmissions of CT DLPDUs.

e) The DLE shall update

— its locally-maintained sense of the link-id of that link, within the extended-link, which is the time source, V(TSL);

— its locally-maintained sense of the local LAS node, V(LN);

— its locally-maintained sense of the quality of DL-time, V(TQ);

— its DL-time-offset, V(DLTO);

— its local-link-scheduling-time-offset, V(LSTO); and

— its node-timer frequency

by comparing the local variables V(TQ), V(TSL) and V(LN) with the corresponding fields of the received TD DLPDU — N_S(TQ) and N_S(TSL) and the DLPDU's source NODE DL-address — to detect changes in either the reference source for the time or in the time distribution path:

1) If the TD DLPDU's source NODE DL-address differs from the receiving DLE's V(LN), then

i) the DLE shall set V(LN) equal to the source NODE DL-address from the received TD DLPDU;

ii) the DLE shall inform local DL-management that a change of time source on the local link has occurred; and

iii) if the DLE requires a measurement of round-trip-delay between itself and the local LAS, to maintain the required accuracy for its time-synchronization class, then the DLE shall invalidate V(MD), the prior round-trip-delay-measurement with the current LAS.

2) If the DLE requires a measurement of round-trip-delay between itself and the local LAS, and if the current value for that round-trip delay, V(MD), is known to be invalid, then the DLE shall form an RQ DLPDU addressed to V(LN).0, and append the DLPDU as specified in 9.4.5, at NORMAL priority to the DLE's NODE DL-address user-request queue, Q_N(UR).

NOTE — This is not the queue normally used with the procedures of 9.4.5.

3) If the measurement of round-trip-delay to the current LAS, V(MD), is valid, then the DLE shall set its V(TQ) from the node-time-quality value, N_S(TQ), in the received DLPDU, after adjustment of the number-of-intervening-links subfield for the one extra intervening link between the sending and receiving DLEs (by incrementing N_S(TQ).LLL when forming V(TQ) to account for the receiving DLE).

NOTE — Therefore a DLE with its own time source, which is not also serving as the source of DL-time to the local link, must ignore its own time source and synchronize with the other DLEs on the local link.

4) If the time-source-link value, N_S(TSL), in the received DLPDU differs from the receiving DLE's V(TSL), and the round-trip-delay-measurement with the current LAS, V(MD), is valid, then the DLE shall

i) set V(TSL) equal to N_S(TSL);

ii) set all other local references to the time-source-link, such as the link-id of the source of time distribution of periodic scheduled activities in Annex B/B.3.5.1(j), equal to N_S(TSL); and

iii) inform local DL-management that a change of time source on the extended link has occurred.

5) If the DL-time-offset, N_S(DLTO), in the received DLPDU differs from the receiving DLE's V(DLTO), then the DLE shall

i) set V(DLTO) equal to N_S(DLTO);

ii) adjust all other local references based on DL-time, such as the periodic schedule DL-time base (T0) in Annex B/B.3.5.1(k), by the amount that V(DLTO) just changed, to reflect the new DL-time base; and

iii) inform local DL-management that a change in V(DLTO) has occurred.

6) If the measurement of round-trip-delay to the current LAS, V(MD), is valid, then the DLE shall compute

$$TEMP = (N_S(DLT) + N_S(DLTA) - N_S(DLTO) - V(LSTO) - N_R(NT) +) \text{ modulo } 2^{56} \qquad \text{(Eq. 20)}$$

NOTE — The above equation is equivalent in form to the computations in (a).

In the following computations, the DLE shall use the value of *K* specified in Table 70.

Table 70 — Time Synchronization Computation

time-synchronization class	value of *K* for use in (e.4)
1 µs	4
10 µs	41

100 µs	410
1 ms	4096
10 ms	40 960
100 ms	409 600
1 s	4 096 000

i) If abs (*TEMP*), the absolute value of *TEMP*, is greater than *K*, and this has not been the case for the prior two received TD DLPDUs, then the DLE shall

A) set V(LSTO) equal to (*TEMP* + V(LSTO)) modulo 2^{56}; and

B) inform local DL-management that a discontinuous change in V(LSTO) has occurred.

ii) If abs(*TEMP*), the absolute value of *TEMP*, is less than or equal to *K*, or if it was greater than *K* for the prior two received TD DLPDUs, then the DLE shall use the computed value, *TEMP*, together with prior computed values as appropriate, to adjust the frequency of incrementation of the DLE's C(NT)

A) to reduce the long-term precession of V(LSTO), relative to

$N_S(DLT) + N_S(DLTA) – N_S(DLTO)$, to zero; and

B) to keep the long-term counting rate of C(NT) to increment by approximately 8 192 000 ($2^{13} \times 10^3$) counts/s.

This frequency adjustment shall be maintained until a newer adjustment is computed.

NOTE — The details of the filtering algorithm which determines this frequency adjustment are intentionally left to the implementor.

9.4.1.3.1 Additional actions required of a bridge

If the TD DLPDU was received at an active port in the forwarding state (see Annex C) which is the bridge DLE's root port (that is, the active port toward the root of the bridge spanning tree), then the DLE shall forward the DLPDU reception event to the bridge operation level (see 5.1.2). The bridge operation level shall share the received sense of time with

— those other active ports of the bridge which are currently serving as the LAS on their own local links;

— all bridge ports which are not in the forwarding or listening states; and

— any local DLS-user.

The following also apply:

a) If the bridge has any port other than its root port which is acting as LAS, then the bridge shall

1) update the DL-time-offset, V(DLTO), for each of those other ports, such that the sum of C(NT), V(LSTO) and V(DLTO) for that non-root port equals the sum of C(NT), V(LSTO) and V(DLTO) for the root port;

NOTE — The DL-time-offset, V(DLTO), and local-link-scheduling-time-offset, V(LSTO), are port-specific (as are most of the variables of 5.7.5, and thus are maintained independently for each port.

2) if (1) caused a change in the DL-time-offset, V(DLTO), then adjust the periodic schedule DL-time base, T0, in B.3.5.1(k) by the same amount that V(DLTO) just changed, to reflect the new DL-time base; and

3) if there was a change in the time-source-link, V(TSL), then set the link-id of the source of time distribution of periodic scheduled activities in B.3.5.1(j), equal to the new value of V(TSL).

b) The bridge operation level may schedule the sending of additional TD DLPDUs on those other ports which are acting as LAS.

c) The contents of the time subfields of those later DLPDUs shall be based on the then-current values for the sending port's internally-maintained node-time variables, C(NT), V(LSTO) and V(DLTO), rather than the value of the DL-time parameters from the just-received TD DLPDU.

d) The contents of the time-source and time-quality subfields — V(TSL) and N(TQ) — of those later DLPDUs shall be based on the corresponding fields from the just-received TD DLPDU, except that the sent N(TQ) shall reflect both

— any impact of the DLE's timeliness class on the resolution of the forwarded time; and

— the fact that there is one additional bridge in the forwarding path.

9.4.1.4 Receipt of an RQ DLPDU

When the DLE receives an RQ DLPDU, it shall form and send as an immediate reply a ROUND-TRIP-DELAY REPLY (RR) DLPDU

a) whose destination DL-address equals the source address of the received RQ DLPDU;

b) whose source address is formed by concatenating the sender's node-id, V_S(TN), and an octet of zero; and

c) whose first two parameter subfields equal the parameter subfields, respectively, of the received and appended-to RQ DLPDU.

9.4.1.5 Receipt of an RR DLPDU

When the DLE receives an RR DLPDU, if the DLE maintains C(NT), then it shall, within some reasonable period not exceeding 60 s, compute the round-trip delay, RTD, as a function of the four parameter subfields of the received and appended-to RR DLPDU, as:

$$\text{RTD} = (\text{ short-time}_4 - \text{short-time}_3 + \text{short-time}_2 - \text{short-time}_1 \text{) modulo } 2^{24} \qquad \text{(Eq. 21)}$$

where the fields are numbered in their order of transmission (or appending) within the DLPDU. If the remote DLE in the measurement was the current LAS DLE, V(LN), then the computed delay shall be saved as the measured-delay

$$\text{V(MD)} = \text{RTD} \qquad \text{(Eq. 22)}$$

which thereby becomes valid for use in C(NT) frequency-adjustment computations, as specified in Eq.15 (see 9.4.1.3).

The results of several round-trip measurements with the same remote node may be combined to reduce the mean error in V(MD).

9.4.1.6 Expiration of the timer T(TDP)

If the timer T(TDP) expires, then

a) if the DLE is the local link's LAS, then the DLE shall note the requirement to transmit a TD DLPDU at the first opportunity;

b) otherwise the DLE shall note the requirement to transmit a CT DLPDU at the first opportunity.

9.4.2 Operation of the compel-service service

The only primitive of the compel-service service is DL-COMPEL-SERVICE request.

9.4.2.1 Receipt of a DL-COMPEL-SERVICE request primitive

When the DLE receives a DL-COMPEL-SERVICE request, it shall classify the request and take the appropriate corresponding action. If the request is for

a) A local (to the DLE) DLSAP-address whose DL(SAP)-role is BASIC, or a peer or publisher or subscriber DLCEP, for which the DL-scheduling-policy is EXPLICIT, then

1) If the DL-address is bound to a sending queue, and the number of DLSDUs in the third partition of that DL-address's user-request queue, $Q_A(UR)$, which are waiting for a DL-COMPEL-SERVICE request primitive is non-zero, then the DLE shall

i) adjust the local DL-address's user-request queue, $Q_A(UR)$, by reducing that number (of DLSDUs in the third partition) by one;

ii) reassign the highest-priority request, from that third partition to the second partition, unless such movement would cause the sending window size of the addressed DLCEP to be exceeded;

NOTES —

1. The restriction in (ii) above applies only to peer and publisher DLCEPs, and never to DLSAP-addresses.

2. Implementations are permitted to defer the actual reassignment until the occurrence of a transmit opportunity for the DLSDU being reassigned. Such deferral may permit a subsequently-requested higher-priority DLSDU to be reassigned in lieu of the DLSDU that would have been reassigned at the moment when the DL-COMPEL-SERVICE request was made. However, in no case may an implementation accumulate a "quota of DL-COMPEL'd requests" which extends beyond the next time that the third partition of the queue $Q_A(UR)$ becomes empty.

iii) form a **reference** to that DL-address's $Q_A(UR)$ at the priority specified in the DL-COMPEL-SERVICE request, where the **reference** indicates the need to send a DLSDU from the sending queue identified in (1), and append the **reference** to either

A) the specified scheduled sequence, if the DL-COMPEL-SERVICE request specified a sequence-identifier; or

B) the DLE's unscheduled-service queue, Q(US), if no Sequence-identifier was specified in the request;

iv) return an immediate status of "success."

Otherwise, when no reassignment occurred, the DLE shall return an immediate status of "user failure — no DLSDU to release."

2) If the DL-address is bound to a sending buffer, then the DLE shall

i) modify the variable $V_{C,K}(SS)$, for the appropriate K corresponding to the DLSDU currently associated with the buffer, to indicate that all segments of the DLSDU require transmission;

ii) form a **reference** to the $Q_A(UR)$ of the specified local peer or publisher DLCEP, at the DLCEP's priority, where the **reference** indicates the need to send a DLSDU from the sending buffer identified in (2), and append the **reference** to either

A) the specified scheduled sequence, if the DL-COMPEL-SERVICE request specified a sequence-identifier; or

B) the DLE's unscheduled-service queue, Q(US), if no Sequence-identifier was specified in the request;

iv) return an immediate status of "success."

b) The remote peer or publisher DLCEP of a local peer or subscriber DLCEP, respectively, then the DLE shall

1) form a **reference** to the Q_A(UR) of the specified local peer or subscriber DLCEP, at the DLCEP's priority, where the **reference** indicates the need to compel the transmission of a DLSDU from the remote correspondent peer or publisher DLCEP identified in (b), and append the **reference** to either

i) the specified scheduled sequence, if the DL-COMPEL-SERVICE request specified a sequence-identifier; or

ii) the DLE's unscheduled-service queue, Q(US), if no sequence-identifier was specified in the request;

NOTE — The above reference is distinguishable from references which result in local release of DLSDUs.

2) return an immediate status of "success;"

c) A local (to the DLE) DLSAP-address whose DL(SAP)-role is INITIATOR, then the DLE shall

1) form a **reference** to that DL-address's Q_A(UR) at the priority specified in the DL-COMPEL-SERVICE request, where the **reference** indicates the need to compel an instance of the unitdata-exchange service with the specified remote DLSAP-address, and append the **reference** to either

i) the specified scheduled sequence, if the DL-COMPEL-SERVICE request specified a sequence-identifier; or

ii) the DLE's unscheduled-service queue, Q(US), if no sequence-identifier was specified in the request;

2) return an immediate status of "success."

d) Some other DL-address, then the DLE shall return an immediate status of "user failure — invalid DL-address."

9.4.3 Operation of the sequence scheduling service

The primitives of the sequence scheduling service are DL-SCHEDULE-SEQUENCE request and confirm, and DL-CANCEL-SCHEDULE request and confirm.

The ability of a DLE to respond positively to a DL-SCHEDULE-SEQUENCE request is determined by both the DLE's own abilities to service a dynamic (that is, not pre-configured by DL-management) request, and the current LAS DLE's ability to support that scheduling process. The conformance claimed in 11.3(c) gives some information on the DLE's capabilities.

9.4.3.1 Receipt of a DL-SCHEDULE-SEQUENCE request primitive

When the DLE receives a DL-SCHEDULE-SEQUENCE request, it shall assign a schedule identifier to the request and return that identifier as part of the DL-SCHEDULE-SEQUENCE request primitive.

Upon completion of the scheduling request, either successfully or after failure, the DLE shall issue a DL-SCHEDULE-SEQUENCE confirm with the same schedule identifier, conveying the status of the scheduling request to the DLS-user. If the scheduling request failed, then the DLE shall release the schedule identifier, making it available for subsequent reuse.

NOTE 1 — DL-SCHEDULE-SEQUENCE confirm indicates completion of the *scheduling* of the associated sequence, either locally or in conjunction with the local LAS; it does not indicate completion of the *execution* of that sequence. An implementation which defers issuing DL-SCHEDULE-SEQUENCE confirm primitives until after the completion of their scheduled sequences is erroneous.

The sequence itself is defined in terms of primitive sequence components specified in 29.3.2.2 of Part 3 of this International Standard, and in Annex B to this part of this International Standard.

NOTE 2 — The specific representations of these primitive sequence components within an end-system is a local-view issue, outside the scope of open system standardization. However, Annex B does include a standardized representation for some of these components when they are communicated between end-systems and an LAS DLE, and implementations may choose to use that same representation locally. Formal DL-programming-interface specifications, which would include the details of such an encoding, could standardize this or a similar local representation.

If the DLE does not implement the remaining-duration counter, C(RD) (see 5.7.1.11), as either an accurate or an approximate counter, then the DLE shall consider all scheduled sequences to be non-interruptable.

The DLE shall determine the class of scheduling operation requested, based on the sequence definition and the type of schedule requested.

If the Schedule Type parameter has the value ONE-TIME or PERIODIC, then

a) if the sequence consists of a single element, and the Schedule Type parameter has the value ONE-TIME, and the Desired Starting Time parameter specifies IMMEDIATE, and the sequence-priority parameter specifies a priority less than or equal to the intrinsic priority of the single element, then the DLE shall append a reference to the requested sequence at the Sequence Priority to the DLE's unscheduled-service queue, Q(US);

b) else if (a) does not apply and the sequence consists of one or more elements, all of which are references to DLCEPs remote from the DLE, none of which are for DLCs specifying EXTRA DLPDU-authentication, then the DLE shall

1) form an SPDU equivalent to the sequence, of the same priority as the sequence;

2) include that SPDU as the DLSDU of a connectionless DT DLPDU of TIME-AVAILABLE priority with a SHORT destination DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2), and source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions;

3) append the DT DLPDU to the DLE's NODE user-request queue, Q_N(UR), at the relevant priority; and

4) append a reference to Q_N(UR) to the DLE's unscheduled-service queue, Q(US);

c) else if (a) and (b) do not apply then the DLE shall

1) use the DLS-user-provided DLSEP-address, or allocate a DLSEP-address if one was not provided;

2) associate the requested sequence to the DLSEP-address, so that receipt of an ES DLPDU specifying that DLSEP-address will cause execution of the associated sequence;

3) compute the maximum uninterrupted duration of the request;

4) form a scheduling-request SPDU with priority equal to the Sequence-Priority of the DLS-user request; where the SPDU requests that the token be delegated to the sequence at the specified DLSEP-address, for the computed duration; and

5) include that SPDU in a DT DLPDU and queue that DLPDU for transmission as specified in (b.2) through (b.4).

If the Schedule Type parameter has the value REPETITIVE, then

d) the DLE shall append a reference to the requested sequence at the Sequence Priority to the DLE's unscheduled-service queue, Q(US).

It is permitted, but not desirable, for a sequence that meets the requirements of (b) to be executed locally as in (a), (c) or (d). It is also permitted, but not desirable, for a sequence that meets the requirements of (a) or (c) to be sent as an SPDU as in (b) to the LAS for remote execution; this permission requires that the SPDU syntax (see Annex B) be capable of conveying the necessary information.

NOTE — Application of a DLS-request confirm delay to a DL-SCHEDULE-SEQUENCE request primitive is for future study.

9.4.3.2 Receipt of a DL-CANCEL-SCHEDULE request primitive

When the DLE receives a DL-CANCEL-SCHEDULE request, it shall determine the class of scheduled sequence being canceled:

a) If the sequence is locally scheduled (see 9.4.3.1(a) or (d)), then the DLE shall

1) remove the reference to the schedule from the unscheduled-service queue, Q(US);

2) issue a DL-CANCEL-SCHEDULE confirm with the identifier of the canceled schedule; and

3) release the schedule identifier, making it available for subsequent reuse;

b) If the sequence is centrally scheduled but locally executed (see 9.4.3.1(c)), then the DLE shall

1) remove the reference to the schedule from the DLSEP-address associated with the schedule;

2) form a cancel-schedule SPDU (see Annex B/B.3.4.4);

3) include that SPDU as the DLSDU of a connectionless DT DLPDU of TIME-AVAILABLE priority with a SHORT destination DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2), and source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions;

4) append the DT DLPDU to the DLE's NODE user-request queue, Q_N(UR), at the relevant priority; and

5) append a reference to Q_N(UR) to the DLE's unscheduled-service queue, Q(US);

c) If the sequence is centrally scheduled and centrally executed (see 9.4.3.1(b)), then the DLE shall form and send a cancel-schedule SPDU as specified in (b.2) through (b.5).

9.4.3.3 Receipt of a schedule-canceled SPDU

If the DLE receives a schedule-canceled SPDU in response to a cancel-schedule SPDU, then the DLE shall

a) issue a DL-CANCEL-SCHEDULE confirm with the identifier of the canceled schedule; and

b) release the schedule identifier and the associated DLSEP-address, if any, making it (them) available for subsequent reuse.

9.4.3.4 LAS-initiated schedule cancellation

If the DLE receives a schedule-canceled SPDU from the LAS, then the DLE shall determine the class of scheduled sequence being canceled:

a) If the sequence is centrally scheduled but locally executed (see 9.4.3.1(c)), then the DLE shall

1) remove the reference to the schedule from the DLSEP-address associated with the schedule;

2) issue a DL-CANCEL-SCHEDULE confirm with the identifier of the canceled schedule; and

3) release the schedule identifier and the associated DLSEP-address, making them available for subsequent reuse.

b) If the sequence is centrally scheduled and centrally executed (see 9.4.3.1(b)), then the DLE shall

1) issue a DL-CANCEL-SCHEDULE confirm with the identifier of the canceled schedule; and

2) release the schedule identifier, making it available for subsequent reuse.

9.4.4 Operation of the sequence subsetting service

The primitives of the sequence subsetting service are DL-SUBSET-SEQUENCE request and confirm.

9.4.4.1 Receipt of a DL-SUBSET-SEQUENCE request primitive

When the DLE receives a DL-SUBSET-SEQUENCE request sequence,

a) if the specified sequence is not defined or not subsettable, then the DLE shall return an appropriate error status and reject the request;

b) otherwise, when (a) does not apply, then the DLE shall subset the sequence as requested and issue a DL-SUBSET-SEQUENCE confirm primitive with the request identifier provided by the DL-SUBSET-SEQUENCE request.

9.4.5 Implicit scheduling of DLS-user requests

When the DLE receives a DLS-user request or response primitive, and the applicable DL-scheduling-policy (see 5.2.6) is IMPLICIT, then the DLE shall execute that request or response as soon as possible, subject to the implicit, or explicit, priority of the request or response. If the request or response requires transmission of one or more DLPDUs, then the DLE shall

a) select the appropriate queue, Q_A(UR), to which the request should be appended:

1) for connectionless-mode procedures (see 9.3), the associated sending DLSAP-address;

2) for connection-oriented procedures (see 9.2),

i) the associated local DLCEP-address for DT DLPDUs when the sending DLCEP's DLCEP-class is PEER or PUBLISHER; or

ii) the associated sending (calling or responding) DLSAP-address for EC, DC and RC DLPDUs; or

iii) either (i) or (ii) for DT DLPDUs when the sending DLCEP's DLCEP-class is SUBSCRIBER;

3) for other procedures (see 9.4), the queue designated by the procedure;

b) append the request, at the appropriate priority, to that Q_A(UR), and adjust that queue, if necessary, so as not to increase the total number of queue members which are in the third partition of the queue, if any; and

c) append a reference to that Q_A(UR), at the appropriate priority, to the DLE's scheduled-service queue, Q(US).

10 Other DLE elements of procedure

10.1 DLE initialization

10.1.1 Hardware or host-system initialization

Upon power-up or after being reset, a DLE shall enter the OFFLINE DL-state, in which it is incapable of transmission on the local link, and shall issue a DLM-EVENT indication notifying DL-management of the event. The DLE shall remain in the OFFLINE state until instructed by DL-management to commence normal DL-operation.

DL-reception in the OFFLINE state is permitted but is not required.

10.1.2 Receipt of a DLM-ACTION request primitive

When the DLE receives a DLM-ACTION request, it shall associate the DLMS-user-specified request identifier with the request. If the request is accepted, as indicated by a returned status of "request accepted" for the DLM-ACTION request, then upon completion of the request, either successfully or after failure, the DLE shall issue a DLM-ACTION confirm with the same request identifier, conveying the status of the request to the DLMS-user.

10.1.2.1 Receipt of a DLM-ACTION request primitive requesting a change in DLE state

Figure 9 shows the state transitions between a DLE's offline and online states.

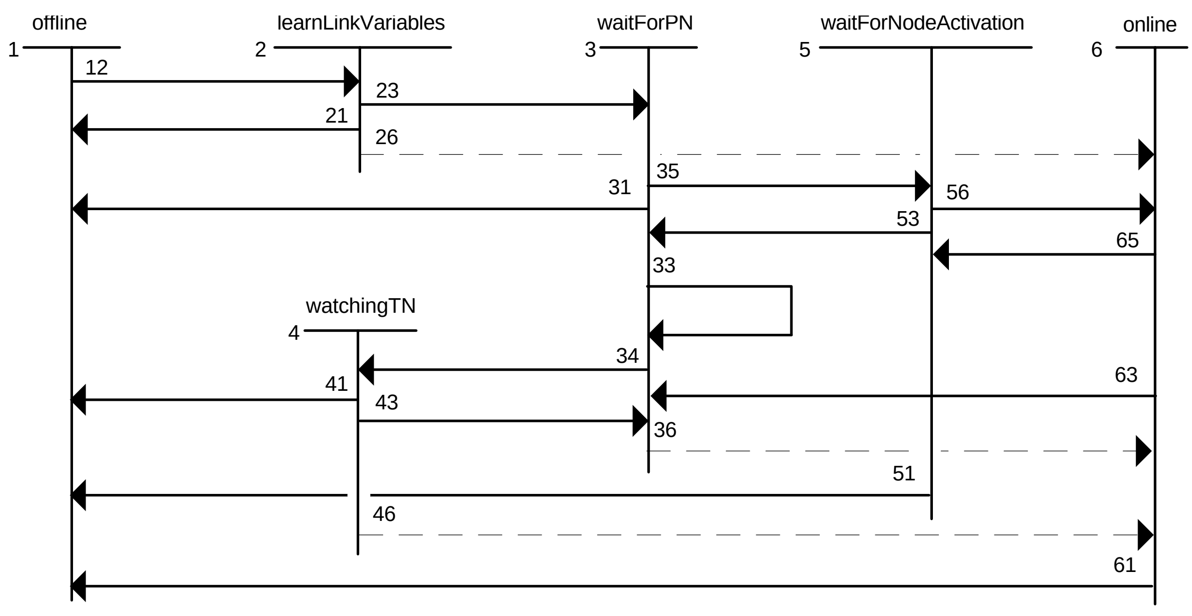


Figure 9 — State transitions of a DLE

If the DLM-ACTION request specifies that the DLE should change its DL-state to ONLINE, then

a) If the DLE's currently-assigned NODE DL-address is in the range $F8_{16}$ .. FF_{16} (see Annex A/A.2.2), then the DLE shall randomly choose the low-order two bits of that NODE DL-address, adopt that modified address as its currently-assigned address, and use the resulting address until a new NODE DL-address is assigned to (or by) the DLE. The actual random choice shall be statistically independent of similar choices made by other DLEs.

NOTE — This requirement for statistical independence minimizes the probability of repeatedly-identical choices by identically-constructed real devices, and thus provides the basis for eventual discrimination among multiple DLEs which happen to choose or respond to the same NODE DL-address, either through fault or misconfiguration.

b) If the DLE is capable of link-master-class or bridge-class operation, and the DLE has knowledge of

— the slot-time, V(ST);

— the maximum-inactivity-to-claim-LAS-delay, V(MICD);

— the Ph-parameters required for forming the PN DLPDU (see 7.13.2); and

— the DL-parameters of the local link enumerated in the node-activation SPDU (see Annex B/ B.3.2.2),

then the DLE shall enable the link-activity-monitoring and CL DLPDU transmission activities of 7.19.3, and if successful in claiming the scheduler token, shall

— change its DL-state to ONLINE; and

— issue a DLM-EVENT indication notifying DL-management of the event.

c) The DLE shall enable the NODE DL-address monitoring and PR DLPDU transmission activities of 7.13.4, 7.15.4 and 7.14.3:

1) The DLE shall learn the DLL protocol version and some of the PhL and DLL configuration parameters of the local link from the PN-parameters of any received PN DLPDU and enter the waitForPN state.

2) The DLE shall monitor the apparent usage of its currently-assigned NODE DL-address (by another DLE on the local link) as follows:

i) If the currently-assigned NODE DL-address is in the range 10_{16} .. $F7_{16}$ (see Annex A/ A.2.2), and the DLE detects that its currently-assigned NODE DL-address is in use by another DLE (by detecting a PT DLPDU addressed to that NODE DL-address and a subsequent immediate-response DLPDU, using the same procedure that the LAS uses to monitor the response to a transmitted PT DLPDU as specified in 7.15.3.2), then the DLE shall first change its NODE DL-address to $F8_{16}$ and then restart this procedure with step (a);

ii) If the currently-assigned NODE DL-address is in the range $F8_{16}$ .. FF_{16} (see Annex A/ A.2.2), and the DLE infers that its currently-assigned NODE DL-address is probably in use by another DLE (by detecting a PT DLPDU addressed to that NODE DL-address), then the DLE shall restart this procedure with step (a);

3) If the DLE receives a PN DLPDU addressed to its currently-assigned NODE DL-address, and

— the DLE is not in the waitForNodeActivation state;

— the DLE supports the protocol version specified in that PN DLPDU;

— the DLE and its associated PhE are configured as specified in that PN DLPDU; and

— the DLE is able to respond within the maximum-reply-delay, V(MRD), specified by the received PN DLPDU,

NOTE — The DLE may have learned the link parameters from an earlier PN DLPDU, but it is necessary to check the parameters of the just received PN DLPDU, and reconfigure if necessary, because the parameters may have changed.

then

i) the DLE shall respond as in 7.13.4 and 7.14.3;

ii) if

— the DLE's required minimum-inter-DLPDU-delay is less than the value of V(MID) specified in the received PN DLPDU; and

— the DLE's maximum-response-delay is less than the value of V(MRD) specified in the received PN DLPDU,

then the DLE shall change its state to waitForNodeActivation; and

iii) if (ii) does not apply, then the DLE shall restart this procedure with step (a).

4) If the DLE is in the waitForNodeActivation state, and

i) if a PN or PT DLPDU is received then the DLE shall restart this procedure with step (a).

ii) if a DT DLPDU is received, and the DT DLPDU contains a node-activation SPDU (see Annex B/B.3.2.2), and the N(RID) field of that SPDU equals the value of V(RID) (see 5.7.1.18) which was used in the probe-response SPDU conveyed in the PR DLPDU last sent by this DLE, then the DLE shall

— change its configuration variables to reflect the contents of that received node-activation SPDU;

— change its DL-state to ONLINE; and

— issue a DLM-EVENT indication notifying DL-management of the event.

iii) if

A) a DT DLPDU with two addresses is received;

B) the source address of the DT DLPDU is the default address for the local-link's LAS; and

C) either the DT DLPDU does not contain a node-activation SPDU, or the N(RID) field of the received node-activation SPDU is not equal to the value of V(RID) (see 5.7.1.18) which was used in the probe-response SPDU conveyed in the PR DLPDU last sent by this DLE,

then the DLE shall infer that its currently-assigned NODE DL-address is in use by another DLE, shall reapply the randomization procedure of (a), if appropriate, shall change its state to waitForPN, and shall behave as in (2) and (3).

If the DLM-ACTION request specifies that the DLE should change its DL-state to OFFLINE, then the DLE shall

— change its DL-state to OFFLINE; and

— issue a DLM-EVENT indication notifying DL-management of the event.

10.1.3 Receipt of a PN DLPDU while in the ONLINE state

If the DLE is not in any of initialization states of 10.1.2.1 and if the DLE receives a PN DLPDU addressed to its currently-assigned NODE DL-address, and the DLE supports the protocol version specified in that PN DLPDU, then

— the DLE and its associated PhE shall be configured as specified in that PN DLPDU; and

— if the DLE is able to respond within the maximum-reply-delay, V(MRD), specified by the received PN DLPDU, then

i) The DLE shall respond as in 7.13.4 and 7.14.3.

ii) If

— the DLE's required minimum-inter-DLPDU-delay is less than the value of V(MID) specified in the received PN DLPDU; and

— the DLE's maximum-response-delay is less than the value of V(MRD) specified in the received PN DLPDU,

then the DLE shall change its state to waitForNodeActivation and behave as specified in 10.1.2.1(c.4).

NOTE — The DLE shall not disconnect any DLCEPs or do any other initialization and continue to operate normally after it changes its state from waitForNodeActivation to ONLINE.

iii) If

— the DLE does not support the protocol version specified in the received PN DLPDU;

— the DLE is not able to respond within the maximum-reply-delay, V(MRD), specified by the received PN DLPDU; or

— ii) does not apply,

then the DLE shall

— issue a DL-Disconnect indication to all non-idle DLCEPs;

— initialize the state of all DLCEPs to Idle;

— reset all queues to empty;

— issue a confirm primitive with an appropriate status for any outstanding DLS-user request;

— perform any other initialization necessary to transition the DLE to the OFFLINE state; and

— start with the procedure of 10.1.2.1(b).

10.2 LAS behavior and operation

10.2.1 LAS operation when holding a scheduler token

The LAS DLE shall initiate each scheduled transaction at its scheduled time within the jitter limits permitted by the schedule for that transaction.

The LAS DLE shall repetitively use the following procedure to determine the next (unscheduled) transaction, subject to the constraint that the transaction can be completed before the next scheduled activity:

a) If the LAS DLE is required to send a TD DLPDU, then it shall send a TD DLPDU.

b) If the LAS DLE has not completed the current cycle of "circulating the token," as indicated by the fact that

— the next DLE to receive the PT DLPDU is not the lowest-numbered NODE DL-address represented in V(LL); or

— the value of the final-use subfield of that next PT DLPDU will be CONTINUE,

then

1) The LAS DLE may use the available bus capacity for link maintenance as specified in 10.2.1(c.2).

2) If the LAS DLE is required to send an ES DLPDU as specified in 10.2.2(b), then it may use the available bus capacity for sending that ES DLPDU as specified in 7.16.3.2, provided this use is limited to sending one ES DLPDU to the DL-address specified in the ES DLPDU per cycle of “circulating the token;”

3) If the available link capacity permits, then the LAS DLE shall send a PT DLPDU as specified in 7.15.3.2;

4) Otherwise, when (3) does not apply and the available link capacity is not sufficient to send a PT DLPDU, the LAS DLE shall repetitively use the following criteria to determine the next DLPDU to be transmitted:

i) If a node-activation SPDU is scheduled for transmission, and there is sufficient available link capacity to send the SPDU in a DT DLPDU as specified in 7.7, then the LAS DLE shall transmit the SPDU within the envelope of a DT DLPDU addressed to the intended recipient.

ii) If (i) does not apply, and there is sufficient available link capacity to probe a new node address as specified in 7.13, then the LAS DLE shall perform the procedure of 7.13.3.

iii) If neither (i) nor (ii) applies, then the LAS DLE may transmit any other DLPDU

— from the generic source address for the local LAS, 0400_{16} (see Annex A/A.3.2);

— which is already queued for transmission,

provided that there is sufficient available link capacity to complete the transaction.

iv) If none of (i) through (iii) applies, and the duration until the next scheduled transaction exceeds token-recovery-delay slot-times, P(TRD) × V(ST) octet-durations, then the LAS DLE shall send one or more IDLE DLPDUs of appropriate length to keep the local link from experiencing an excessive period of inactivity.

Any such link capacity which is not adequate for the procedure of 7.15.3 need not be attributed to the duration of token usage allocated for the LAS DLE’s link-maintenance activities.

c) Otherwise, when (b) does not apply, so that the LAS DLE has just completed a cycle of “circulating the token,” then

1) If the LAS DLE has a need to transfer its role to another DLE, then

i) the LAS DLE shall not start the next cycle of “circulating the token;” but instead

ii) the LAS DLE shall send all DLPDUs which it has already scheduled for transmission from the generic source address for the local LAS, 040016 (see Annex A/A.3.2), after which

iii) the LAS DLE shall send a TL DLPDU as specified in 7.20.3.

2) Otherwise, when (1) does not apply, then the LAS DLE shall send all link-maintenance DLPDUs (see 5.7.5.9) which are queued for transmission, if necessary over an interval of one or more cycles of “circulating the token,” so that the link capacity used for link maintenance during one cycle of “circulating the token” does not exceed the configured link-maintenance-token-holding-time, V(LTHT).

The LAS DLE shall repetitively use the following criteria to determine the next transaction:

i) If there is a need to send a node-activation SPDU, then the DLE shall send such an SPDU (in a DT DLPDU).

ii) If (i) does not apply, and the LAS DLE has not sent any PN DLPDUs during the just completed cycle of “circulating the token,” then the LAS DLE shall send a PN DLPDU to the next NODE DL-address to be probed, as specified in 7.13.3.

iii) If neither (i) nor (ii) applies, then the LAS DLE may send any DLPDU, other than a TD, PN, PT or ES DLPDU, which it needs to send in its role as LAS.

After completing any such permitted transactions, the LAS DLE shall start a new cycle of “circulating the token.”

NOTE — The LAS DLE will perform as many transactions as possible, as specified by (a), (b) and (c), provided that the transactions can be completed before the next scheduled activity.

10.2.2 Return of a delegated token; assumption of a scheduler token

When a delegated token which was delegated by a PT DLPDU is returned by an RI DLPDU, then the LAS DLE shall associate the value of the DD-parameter of that received RI DLPDU with the DLE returning the token, and use it as the minimum token delegation time for sending the next PT DLPDU to that DLE as required in 7.15.3.1(d.3).

When a delegated token which was delegated by an ES DLPDU is returned by an RI DLPDU, then the LAS DLE shall check whether the delegating ES DLPDU was the last delegation of a scheduled sequence, as indicated by the encoding specified in Annex B/B.3.5.2.2(a):

a) If it was not the last delegation within the current cycle of that scheduled sequence execution, then the LAS DLE shall ignore the request for an additional delegation which is requested by the just-received RI DLPDU;

b) Otherwise, when it was the last delegation of a scheduled sequence, then the DLE may, but need not,

1) associate the value of the DD-parameter of that received RI DLPDU with the destination DL-address of that ES DLPDU; and

2) note the need to send another ES DLPDU with a final token use subfield equal to CONTINUE to the requesting DL-address, during an interval of otherwise-unscheduled bus capacity.

When the LAS DLE assumes a scheduler token due to link inactivity, on expiration of a link monitoring interval, then the LAS DLE shall initiate transmission of its next DLPDU such that there is no more than 14 slot-times, $14 \times V(ST)$ octet-durations, of inactivity on the local link.

The LAS DLE may measure the duration for which the token has been delegated, and if that duration expires before the token is returned, then the LAS DLE may assume the scheduler token immediately upon that expiration.

If the delegated token has not been returned, and it is time for the next scheduled activity, then the LAS DLE may assume the scheduler token immediately.

10.2.3 Receipt of a probe-response (PR) SPDU

If one or more PR DLPDUs are received in response to a PN DLPDU sent by the LAS DLE, then the LAS DLE shall process the first received PR DLPDU (see Annex B/B.3.21) as specified in this clause, and shall discard all of the other PR DLPDUs received during that response window.

The LAS DLE shall compare the values specified in the minimum-inter-DLPDU-delay, V(MID), and maximum-response-delay-in-octets, $V(MRD) \times V(ST)$, fields of a received PR SPDU with the same parameters as configured for the local link.

If

— the responding DLE’s V(MID) is less than or equal to the link’s configured value for V(MID); and

— the responding DLE’s V(MRD) is less than or equal to the link’s configured value for V(MRD),

then

a) The LAS DLE shall form a node-activation SPDU with its V(RID) field equal to the V(RID) field of the received PR SPDU.

b) The LAS DLE shall note the need to send a node-activation SPDU to the responding DLE, where that node-activation SPDU is conveyed in a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address equal to the NODE.0 DL-address of the responding DLE's DL-support functions;

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required node-activation SPDU as the conveyed DLSDU.

c) After sending this DT DLPDU, the LAS DLE shall include the NODE DL-address of the responding DLE in its local-link live-list, V(LL).

d) If the token-circulation-not-needed field, N, of the responding PR SPDU was set to 0 (no, token circulation is needed), then the LAS DLE shall include the NODE DL-address of the responding DLE in its local-link-token-circulation-list, V(TCL).

e) The LAS DLE shall publish the live-list change in one of the following two ways:

1) the LAS DLE shall

i) increment the live-list revision-number;

ii) form a live-list-change SPDU (see Annex B/B.3.2.4) indicating the NODE DL-address of the affected DLE and its status; and

iii) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address equal to the group DL-address for the DL-support functions of all LM DLE's on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required live-list-change SPDU as the conveyed DLSDU; or

2) the LAS DLE shall append the NODE DL-address of the affected DLE and its status to an existing DT DLPDU conveying a live-list-change SPDU, formed as in (1) and (2), which has previously been scheduled for transmission but has not yet been sent.

Otherwise, when either

— the responding DLE's V(MID) is greater than the link's configured value for V(MID); or

— the responding DLE's V(MRD) is greater than the link's configured value for V(MRD),

then the DLE shall notify its local DL-management.

10.2.4 Lack of response to a PT DLPDU

If a DLE does not send any DLPDU in response to a PT DLPDU addressed to the DLE, as specified in 7.15.4.1, then the LAS DLE shall note the event. After three such non-responses for any specific NODE DL-address within three consecutive cycles of “circulating the token,” the LAS DLE shall

a) remove that NODE DL-address from the LAS DLE’s local-link token-circulation-list, V(TCL), and local-link live-list, V(LL); and

b) publish the live-list change as specified in 10.2.3(e).

10.2.5 Receipt of a live-list-request SPDU

The LAS shall

a) form a live-list-detail SPDU (see Annex B/B.3.2.6); and

b) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address equal to the source DL-address of the DLPDU which conveyed the live-list-request SPDU, or equal to the group DL-address for the DL-support functions of all LM DLE’s on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required live-list-detail SPDU as the conveyed DLSDU.

10.2.6 Receipt of a relinquish-LAS-role-request SPDU

If the LAS had already noted the need to transfer the LAS role, then the LAS shall ignore the received SPDU, else the LAS shall note the need to transfer the LAS role.

10.2.7 Other link-maintenance requirements

Whenever there is a change in the LAS DLE’s schedule, or whenever an LM DLE other than the most recent LAS DLE assumes the scheduler role, or periodically such that the time between two successive distributions of the LAS database by means of LAS-data-base-status SPDUs is equal to or more than the LAS-data-base-distribution-period, V(LDDP) (see 5.7.5.20) and less than $1.5 \times$ V(LDDP), then the LAS shall

a) form an LAS-database-status SPDU (see Annex B/B.3.2.3); and

b) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address equal to the group DL-address for the DL-support functions of all LM DLE’s on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required LAS-database-status SPDU as the conveyed DLSDU.

10.2.8 Receipt of a link-master-parameters-request SPDU

The LAS shall

a) form a link-master-parameters-reply SPDU (see Annex B/B.3.2.12); and

b) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address equal to the source DL-address of the DLPDU which conveyed the link-master-parameters-request SPDU, or equal to the group DL-address for the DL-support functions of all LM DLE's on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required link-master-parameters-reply SPDU as the conveyed DLSDU.

10.2.9 Receipt of a token-hold-time-request SPDU

If the starting NODE DL-address field of the received SPDU is 00, then the LAS DLE shall

a) form as many token-hold-time-array SPDUs (see Annex B/B.3.2.14) as necessary to send the maximum-token-hold-time for all DLEs represented in the local-link-live-list, V(LL);

b) form one token-hold-time-array SPDU with a starting NODE DL-address field with a value of zero; and

c) for each token-hold-time-array SPDU, schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address equal to the source DL-address of the DLPDU which conveyed the token-hold-time-request SPDU, or equal to the group DL-address for the DL-support functions of all LM DLE's on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required token-hold-time-array SPDU as the conveyed DLSDU;

such that the token-hold-time-array SPDU with a starting NODE DL-address field with a value of zero is sent after all of the other token-hold-time-array SPDUs have been sent.

Otherwise, when the starting NODE DL-address field of the received SPDU is $N0_{16}$, where $N = 1$ to F_{16}, then the LAS DLE shall

d) form one token-hold-time-array SPDU with a starting NODE DL-address field equal to the starting NODE DL-address field of the received SPDU; and

e) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address equal to the source DL-address of the DLPDU which conveyed the token-hold-time-request SPDU, or equal to the group DL-address for the DL-support functions of all LM DLE's on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required token-hold-time-array SPDU as the conveyed DLSDU.

Else, the received SPDU is invalid and shall be ignored.

10.2.10 Receipt of a schedule-summary-request SPDU

The LAS shall

a) form a schedule-summary SPDU (see Annex B/B.3.5.1); and

b) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— TIME-AVAILABLE priority;

— a SHORT destination DL-address equal to the source DL-address of the DLPDU which conveyed the schedule-summary-request SPDU, or equal to the group DL-address for the DL-support functions of all LM DLE's on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required schedule-summary SPDU as the conveyed DLSDU.

10.2.11 Receipt of a sub-schedule-request SPDU

If the schedule version-number field of the received SPDU is equal to the schedule version-number field of the currently active schedule, then

a) If the sub-schedule identifier field of the received SPDU is 00, then the LAS DLE shall

1) form as many sub-schedule SPDUs (see Annex B/B.3.5.2) as necessary to send all of the sub-schedules of the currently active schedule;

2) form one sub-schedule SPDU with a sub-schedule identifier field with a value of zero; and

3) for each sub-schedule SPDU, schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— TIME-AVAILABLE priority;

— a SHORT destination DL-address equal to source DL-address of the DLPDU which conveyed the sub-schedule-request SPDU, or equal to the group DL-address for the DL-support functions of all LM DLE's on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required sub-schedule SPDU as the conveyed DLSDU;

such that the sub-schedule SPDU with sub-schedule identifier field equal zero is sent after all other sub-schedule SPDUs are sent.

b) Otherwise, when the sub-schedule identifier field of the received SPDU is non-zero and equal to the sub-schedule identifier of one of the sub-schedules of the currently active schedule, then the LAS DLE shall

1) form one sub-schedule SPDU with a sub-schedule identifier field whose value is equal to the value of the sub-schedule identifier field of the received SPDU; and

2) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— TIME-AVAILABLE priority;

— a SHORT destination DL-address equal to source DL-address of the DLPDU which conveyed the sub-schedule-request SPDU, or equal to the group DL-address for the DL-support functions of all LM DLE's on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the required sub-schedule SPDU as the conveyed DLSDU.

c) Otherwise, when the sub-schedule identifier field of the received SPDU is non-zero and is not equal to the sub-schedule identifier of any of the sub-schedules of the currently active schedule, then the LAS DLE shall

1) form a schedule-summary SPDU (see Annex B/B.3.5.1); and

2) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— TIME-AVAILABLE priority;

— a SHORT destination DL-address equal to the source DL-address of the DLPDU which conveyed the sub-schedule-request SPDU, or equal to the group DL-address for the DL-support functions of all LM DLE's on the local link, 0101_{16} (see Annex A/A.3.2);

— a SHORT source DL-address of the local LAS, 0400_{16} (see Annex A/A.3.2);

— a null SD-parameter field; and

— the schedule-summary SPDU as the conveyed DLSDU.

If the schedule version-number field of the received SPDU is not equal to the schedule version-number field of the currently active schedule, then the LAS DLE shall send a schedule-summary SPDU as in (c).

10.3 DL-support operation

A DLE's DL-support functions receive each DT DLPDU with the destination DL-address of NODE.0, V(TN).0 (see 7.7.4.1.1(d)), addressed to that DLE. When the DLE receives such a DLPDU, the DLS-user-data included in the received DT DLPDU shall be interpreted as an SPDU as specified in Annex B:

a) If the received SPDU is a DL-address query SPDU, then the DLE shall process it as specified in 9.2.3.3 or 9.3.4.3 depending upon the DL-address in the SPDU.

b) If the received SPDU is a DL-address-report SPDU, then the DLE shall process it as specified in 9.2.3.4 or 9.3.4.4 depending upon the DL-address in the SPDU.

c) If the received SPDU is a Schedule-canceled SPDU, then the DLE shall process it as specified in 9.4.3.3 or 9.4.3.4.

d) If the source address of the received DT DLPDU is the default address for the local-link's LAS, 4.0, and if the received SPDU is a node-activation SPDU, then the DLE shall process it as specified in 10.1.2.1.

e) If the received SPDU is an LAS-database-status SPDU, then the DLE shall process it as specified in 10.3.1.

f) If the received SPDU is a live-list-change SPDU, then the DLE shall process it as specified in 10.3.2.

g) If the received SPDU is a live-list-detail SPDU, then the DLE shall process it as specified in 10.3.3.

h) If the received SPDU is a live-list-request SPDU, then the DLE shall process it as specified in 10.2.5.

j) If the received SPDU is a relinquish-LAS-role SPDU, then the DLE shall process it as specified in 10.2.6.

k) If the received SPDU is a link-master-parameters-reply SPDU, then the DLE shall process it as specified in 10.3.5.

m) If the received SPDU is a token-hold-time-array SPDU, then the DLE shall process it as specified in 10.3.6.

n) If the received SPDU is a link-master-parameters-request SPDU, then the DLE shall process it as specified in 10.2.8.

p) If the received SPDU is a token-hold-time-request SPDU, then the DLE shall process it as specified in 10.2.9.

q) If the received SPDU is a schedule-summary-request SPDU, then the DLE shall process it as specified in 10.2.10.

r) If the received SPDU is a sub-schedule-request SPDU, then the DLE shall process it as specified in 10.2.11.

s) If the received SPDU is a schedule-summary SPDU, then the DLE shall process it as specified in 10.3.7.

t) If the received SPDU is a sub-schedule SPDU, then the DLE shall process it as specified in 10.3.8.

10.3.1 Receipt of an LAS-database-status SPDU by an LM DLE

The receiving LM DLE shall compare the live-list revision-number specified in the received SPDU with the revision-number of its own local copy of the live-list, V(LL). If the two revision-numbers differ, then the LM DLE shall

a) form a live-list-request SPDU (see Annex B/B.3.2.5);

b) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address for the local LAS, 0400_{16} (see Annex A/A.3.2);

— a SHORT source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions;

— a null SD-parameter field; and

— a live-list-request SPDU as the conveyed DLSDU.

A reference to the DLE's NODE user-request queue, Q_N(UR), shall be appended to the LM DLE's unscheduled-service queue, Q(US).

The receiving LM DLE shall compare the schedule version-number specified in the received SPDU with the schedule version-number of its own local copy of the link schedule. If the two schedule version-numbers differ, and if the schedule-type field, T, in the received SPDU indicates that the LAS DLE is capable of transferring its schedule to other LM DLEs, then the LM DLE shall

c) form a schedule-summary-request SPDU;

d) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address for the local LAS, 0400_{16} (see Annex A/A.3.2);

— a SHORT source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions;

— a null SD-parameter field; and

— a schedule-summary-request SPDU as the conveyed DLSDU.

A reference to the DLE's NODE user-request queue, Q_N(UR), shall be appended to the LM DLE's unscheduled-service queue, Q(US).

10.3.2 Receipt of a live-list-change SPDU by an LM DLE

If a received live-list-change SPDU (see Annex B/B.3.2.4) indicates a change in the live-list, then the receiving LM DLE shall update its own local version of the live-list in the manner indicated by the received SPDU, and shall increment its live-list revision-number.

10.3.3 Receipt of a live-list-detail SPDU by an LM DLE

The receiving LM DLE shall update

— its own local version of the local-link-live-list, V(LL);

— the live-list revision-number;

— the LAS DLE's NODE DL-address, V(LN);

— the expected-non-response-list, V(ENRL);

— the first-unpolled-node's NODE DL-address , V(FUN); and

— the number-of-consecutive-unpolled-nodes, V(NUN) as indicated by the received SPDU.

10.3.4 Request for LAS parameters by an LM DLE

A link master requires the following parameters to operate as the LAS:

a) the local link's configuration parameters defined in 5.7.5 and included in the link-master-parameters-reply SPDU (see Annex B/B.3.2.12);

b) the local-link-live-list, V(LL);

c) the expected-non-response-list, V(ENRL);

d) the token-circulation-list, V(TCL);

e) the maximum-token-holding-time-array, V(MTHA);

f) the local link's schedule; and

g) a sense of the current LAS DLE's DL-time.

If a link master DLE does not have the necessary parameters required to operate as the LAS, as may be the case when the link master DLE has just changed its state to ONLINE, then the link master DLE shall schedule the transmission of the following SPDUs to the local LAS in connectionless DT DLPDUs each with format 1S, NORMAL priority, a destination address of the local LAS, 0400_{16} (see Annex A/A.3.2), and a source address of V(TN).00:

— a link-master-parameters-request SPDU (see Annex B/B.3.2.11);

— a token-hold-time-request SPDU (see Annex B/B.3.2.13);

— a live-list-request SPDU (see Annex B/B.3.2.5); and

— a schedule-summary-request SPDU (see Annex B/B.3.5.3); and

the link-master DLE shall note the need to transmit a CT DLPDU at its first opportunity.

After sending the SPDU, the sending DLE shall wait for reception of the link-master-parameters-reply SPDU, token-hold-time-array SPDUs, live-list-detail SPDU and schedule-summary SPDU;

If these SPDUs are not received by the requesting LM DLE within fifteen receptions of a PT DLPDU with a token-use-subfield equal to RESTART addressed to the DLE's NODE DL-address, then the requesting LM DLE shall again schedule the transmission of those request SPDUs appropriate for the missing reply SPDUs.

10.3.5 Receipt of a link-master-parameter-reply SPDU by an LM DLE

The DLE shall use the received SPDU to update the parameters required to operate as LAS and

— if the DLE's maximum-inactivity-to-claim-LAS-delay is less than or equal to the link's configured value of maximum-inactivity-to-claim-LAS-delay, V(MICD), then the DLE shall note that it is capable of operating as the local link's LAS.

10.3.6 Receipt of a token-hold-time-array SPDU by an LM DLE

If the starting NODE DL-address field of the received SPDU (see Annex B/B.3.2.14) is $N0_{16}$, where N = 1 to F_{16} then the DLE shall

— update the maximum-token-holding-time-array, V(MTHA) from the maximum-token-hold-time values received in the SPDU; and

— exclude from the token-circulation-list, V(TCL), those DLEs for which the maximum-token-hold-time is zero.

Otherwise, when the starting NODE DL-address field of the received SPDU is 00, then the DLE shall

— check whether it has received the token-hold-time-array SPDUs for all NODE DL-addresses in the local-link-live-list, V(LL); and

— if not, then send one or more token-hold-time-request SPDUs, specifying such values for the starting NODE DL-addresses in those SPDUs, that the corresponding reply SPDUs will contain the maximum-token-hold-times for all the NODE DL-addresses in local-link-live-list, V(LL), whose maximum-token-hold-times are not yet known by the requesting DLE;

Otherwise, when the starting NODE DL-address field of the received SPDU is of the form NM_{16}. where M is non-zero, then the received SPDU is invalid and shall be discarded.

10.3.7 Receipt of a schedule-summary SPDU by an LM DLE

If the schedule version-number field of the received SPDU (see Annex B/B.3.5.1) is non-zero, then

a) If either the DLE has no link schedule, or the schedule version-number of the link schedule stored in the DLE is different than the schedule version-number field of the just received SPDU, then the DLE shall

1) check that it is capable of executing the link schedule whose summary was just received. The DLE shall

i) check that it has the capability to execute the number of sub-schedules in the schedule-summary;

ii) check that the DLE's value of maximum-scheduling-overhead, V(MSO), (see 5.7.4.6) is less than or equal to the value of V(MSO) specified in the schedule-summary;

iii) check that the DLE has the required storage capacity specified in the schedule-summary;

iv) check that the DLE has the required timing resolution specified in the schedule-summary;

v) check that the DLE's value of time-source-link, V(TSL), (see 5.7.1.21) is the same as the value of V(TSL) specified in the schedule-summary.

If any of these checks fail, then the DLE shall discard the just received schedule-summary SPDU.

2) otherwise, when these checks are all passed, then the DLE shall

i) form a sub-schedule-request SPDU with its schedule version-number field equal to the value of the schedule version-number field of the just received schedule-summary SPDU, and with a sub-schedule identifier equal to zero; and

ii) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address for the local LAS, 0400_{16} (see Annex A/A.3.2);

— a SHORT source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions;

— a null SD-parameter field; and

— the just-formed sub-schedule-request SPDU as the conveyed DLSDU.

b) Otherwise, when the DLE has a link schedule and the version-number of the link schedule stored in the DLE is equal to the schedule version-number field of the just received SPDU, but the DLE does not have the sub-schedule for all of the sub-schedules identified in the just received schedule-summary SPDU, then the DLE shall

1) form one or more sub-schedule-request SPDUs, each with a schedule version-number field equal to the value of the schedule version-number field of the just received SPDU, and with a sub-schedule identifier equal to one of the sub-schedule-SPDU references of the missing sub-schedules; and

2) schedule the transmission of a connectionless DT DLPDU for each such just-formed SPDU, with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address for the local LAS, 0400_{16} (see Annex A/A.3.2);

— a SHORT source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions;

— a null SD-parameter field; and

— one of the just-formed sub-schedule-request SPDUs as the conveyed DLSDU.

Otherwise, when the schedule version-number field of the just received SPDU is zero, then the DLE shall record that there is no active link schedule.

10.3.8 Receipt of a sub-schedule SPDU by an LM DLE

If the schedule version-number field of the just received sub-schedule SPDU (see Annex B/B.3.5.2) is equal to the schedule version-number field of the last received schedule-summary SPDU (see Annex B/B.3.5.1), then

a) if the sub-schedule identifier field of the received SPDU is equal to a sub-schedule-SPDU reference included in the last received schedule-summary SPDU, then the DLE shall store the sub-schedule as part of the link schedule;

b) otherwise, when the sub-schedule identifier field of the received SPDU is equal to zero, then the DLE shall

— check whether it has received the sub-schedule SPDUs for all of the sub-schedule-SPDU references in the last-received schedule-summary SPDU; and

— if not, then send one or more sub-schedule-request SPDUs, each with a value of the sub-schedule identifier requesting one of the missing sub-schedules;

c) otherwise, when neither (a) nor (b) apply, then the DLE has received a sub-schedule SPDU inconsistent with the DLE's last-received schedule-summary SPDU. Therefore the DLE shall

1) form a schedule-summary-request SPDU; and

2) schedule the transmission of a connectionless DT DLPDU with

— format 1S;

— NORMAL priority;

— a SHORT destination DL-address for the local LAS, 0400_{16} (see Annex A/A.3.2);

— a SHORT source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions;

— a null SD-parameter field; and

— a schedule-summary-request SPDU as the conveyed DLSDU.

Otherwise, when the schedule version-number field of the just received sub-schedule SPDU is not equal to the schedule version-number field of the last received schedule-summary SPDU, or if the DLE had never received a schedule-summary SPDU, then the DLE shall send a schedule-summary-request SPDU as in (c).

10.3.9 Request for LAS transfer by an LM DLE

If an LM DLE receives a request from local DL-management to assume the LAS role, then

— if the DLE does not have the necessary parameters required to operate as the LAS, then it shall follow the procedure of 10.3.4 to obtain those parameters.

After receiving the necessary parameters to operate as LAS, the LM DLE shall

a) form a Relinquish-LAS-role SPDU;

b) schedule the transmission of a connectionless DT DLPDU with

   — format 1S;

   — NORMAL priority;

   — a SHORT destination DL-address for the local LAS, 0400_{16} (see Annex A/A.3.2);

   — a SHORT source DL-address equal to the NODE.0 DL-address, V(TN).0, of the sending DLE's DL-support functions;

   — a null SD-parameter field; and

   — a schedule-summary-request SPDU as the conveyed DLSDU;

c) wait for reception of a TL DLPDU;

d) if a TL DLPDU is not received within fifteen receptions of a PT DLPDU with a token-use-subfield equal to RESTART, then the LM DLE shall retry this process commencing with step (a);

e) if a TL DLPDU is received, but is rejected as specified in 7.20.4.2, then the LM DLE shall assume that it does not have the necessary parameters required to operate as the LAS and shall repeat the procedures of 10.3.9;

f) if a TL DLPDU is still not received after two retry attempts (as in (d) or (e)), then the LM DLE shall inform local DL-management of this failure.

11 Implementation profiles

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

11.1 DL-address, queue and buffer management services

The first profile categories specifies whether an implementation supports LONG DL-addresses (see 6.2.1.1) and thus supports direct communication and direct management through a bridge (see 3.3.4).

Permitted choices are:

YES LONG, SHORT and VERY-SHORT address representations are supported (that is, are recognized and can be generated);

NO only SHORT and VERY-short address representations are supported (that is, are recognized and can be generated).

11.2 DL-data delivery services

The first profile category specifies the degree to which an implementation supports those communications priorities which facilitate preemptive use of the shared communications resource. The remaining profile categories specify the degree to which an implementation supports those DLC data-delivery features which enhance the unaugmented DL-capability of conveying limited amounts of DLS-user information, with limited QoS, between DLSAPs:

a) Three priorities are defined in the standard for both connection-mode and connectionless data communications. The utility for time-critical communications of an implementation of this standard depends substantially on whether the implementation supports multiple priorities of data delivery to the DLS-user. Two levels of support, of decreasing difficulty and utility, have been specified. Which of them does the implementation support?

Permitted choices are:

MULTI URGENT, NORMAL and TIME-AVAILABLE priorities are all supported for both connection-mode and connectionless data communications;

SINGLE only TIME-AVAILABLE priority is supported; URGENT and NORMAL priorities are not supported.

b) Four classes of DLC data delivery features are defined in the standard for both peer and multi-peer DLCs. Of these, three (CLASSICAL, ORDERED and UNORDERED) provide substantively different qualities of service. In addition, the quality of service of CLASSICAL peer-to-peer data delivery is different from that of it analogous multi-peer counterpart.

NOTE — In the absence of communications errors, DISORDERED and CLASSICAL data delivery services are indistinguishable. Thus the DISORDERED data delivery service is an implementation option when CLASSICAL service is provided, and does not provide a substantively different quality of service from the CLASSICAL service.

The utility for time-critical communications of an implementation of this standard depends heavily on the degree to which the implementation supports the data delivery paradigms needed by the DLS-user. Four levels of support, of decreasing difficulty and utility, have been specified. Which of them does the implementation support?

Permitted choices are:

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

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

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

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

c) Multi-peer DLCs can be capable of conveying DLSDUs from subscribers to their publisher, providing a capability roughly equivalent to the inverse of a connectionless multi-cast. Is this optional capability supported?

Permitted choices are:

YES DLSDUs may be conveyed from subscribers to their publisher;

NO DLSDUs cannot be conveyed from subscribers to their publisher.

d) CLASSICAL, ORDERED and DISORDERED peer and publisher DLCs are capable of conveying DLSDUs up to 16 times the size of the DLSDUs conveyable with UNORDERED DLCs or connectionless services. What is the ratio of the maximum size DLSDU conveyable on CLASSICAL and ORDERED peer and multi-peer DLCs to that conveyable with UNORDERED DLCs or connectionless services?

Permitted choices are the integers 1 to 16, inclusive. If CLASSICAL DLCs are supported and the ratios are different for CLASSICAL and ORDERED DLCs, or if the ratios are different for peer and multi-peer DLCs, then the smallest of the ratios shall be specified. If the choice specified for category (b) above was **D**, then this value shall be 1 (one).

e) ORDERED peer and multi-peer DLCs are capable of computing TRANSPARENT, RESIDENCE, UPDATE and SYNCHRONIZED timeliness of DLSDU conveyance. Which of these forms of timeliness computations are supported?

Permitted choices are NONE, just TRANSPARENT, or TRANSPARENT plus any combination of the other three. If the choice specified for category (b) above was **D**, then this value shall be NONE.

f) ORDERED peer and multi-peer DLCs are capable of conveying DL-time stamps with DLSDUs. Are this capability, and the capability of noting the DL-time of DL-PUT request primitives, and of providing TRANSPARENT timeliness, all supported?

Permitted choices are:

YES all three capabilities are supported, with best achievable time resolution dependent on the time class specified in 11.3(a);

NO not all of these capabilities are supported.

If the choice specified for category (b) above was **D**, then this value shall be NO.

g) Three forms of connectionless data delivery are defined in the standard:

1) unitdata transfer with local confirmation;

2) unitdata transfer with remote confirmation; and

3) unitdata exchange.

Which of these forms of connectionless data delivery are supported?

Permitted choices are:

X all three forms are supported;

Y both forms of unitdata transfer are supported; unitdata exchange is not supported;

Z only unitdata transfer with local confirmation is supported; unitdata transfer with remote confirmation and unitdata exchange are not supported;

11.3 DL-time and time-based scheduling services

The last three profile categories specify the degree to which an implementation supports a Fieldbus-wide sense of time and time-based scheduling:

a) The maximum asynchronism in the Fieldbus-shared sense of time determines the coarseness or fineness of such shared activities as distributed time-based scheduling and distributed sequence-of-events determination. Eight classes of time synchronism have been defined:

1 µs Under steady-state conditions, the DL-implementation can maintain a sense of DL-time whose maximum long-term mean error relative to the local LAS's sense of DL-time is at most 250 ns for Basic and Link Master DLEs, and 125 ns for Bridge DLEs, under the condition that a single TD DLPDU is received once each 5 ms, as shown in Table 71.

NOTE — When connected to a similar reference node by a multi-link Fieldbus with no more than two intervening bridges, in which the reference node, the intervening bridges, and the node under measurement all meet this requirement, such a DL-implementation can provide a sense of time differing from that in the reference node by less than 500 ns, and thus from any node similarly removed from the reference node by less than 1 µs.

10 µs Under steady-state conditions, the DL-implementation can maintain a sense of DL-time whose maximum long-term mean error relative to the local LAS's sense of DL-time is at most 3 µs for Basic and Link Master DLEs, and 1 µs for Bridge DLEs, under the condition that a single TD DLPDU is received once each 50 ms, as shown in Table 71.

NOTE — When connected to a similar reference node by a multi-link Fieldbus with no more than two intervening bridges, in which the reference node, the intervening bridges, and the node under measurement all meet this requirement, such a DL-implementation can provide a sense of time differing from that in the reference node by less than 5 µs, and thus from any node similarly removed from the reference node by less than 10 µs.

100 µs Under steady-state conditions, the DL-implementation can maintain a sense of DL-time whose maximum long-term mean error relative to the local LAS's sense of DL-time is at most 25 µs for Basic and Link Master DLEs, and 10 µs for Bridge DLEs, under the condition that a single TD DLPDU is received once each 500 ms, as shown in Table 71.

NOTE — When connected to a similar reference node by a multi-link Fieldbus with no more than two intervening bridges, in which the reference node, the intervening bridges, and the node under measurement all meet this requirement, such a DL-implementation can provide a sense of time differing from that in the reference node by less than 50 µs, and thus from any node similarly removed from the reference node by less than 10 µs.

1 ms Under steady-state conditions, the DL-implementation can maintain a sense of DL-time whose maximum long-term mean error relative to the local LAS's sense of DL-time is at most ms for Basic and Link Master DLEs, and ms for Bridge DLEs, under the condition that a single TD DLPDU is received once each 5 s, as shown in Table 71.

NOTE — When connected to a reference node by a multi-link Fieldbus, in which

a) the reference node, two bridges in series, and the node under measurement all meet this requirement; or

b) the reference node and three bridges in series meet the requirements of the 10 µs class, and an additional two bridges in series and the node under measurement all meet the requirement of the 1 ms class,

such a DL-implementation can provide a sense of time differing from that in the reference node by less than 0,5 ms, and thus from any node similarly removed from the reference node by less than 1 ms.

10 ms Under steady-state conditions, the DL-implementation can maintain a sense of DL-time whose maximum long-term mean error relative to the local LAS's sense of DL-time is at most 2,5 ms for Basic and Link Master DLEs, and 0,5 ms for Bridge DLEs, under the condition that a single TD DLPDU is received once each 10 s, as shown in Table 71.

NOTE — When connected to a reference node by a multi-link Fieldbus, in which

a) the reference node, two bridges in series, and the node under measurement all meet this requirement; or

b) the reference node and five bridges in series meet the requirements of the 1 ms class, and an additional two bridges in series and the node under measurement all meet the requirement of the 10 ms class,

such a DL-implementation can provide a sense of time differing from that in the reference node by less than 5 ms, and thus from any node similarly removed from the reference node by less than 10 ms.

100 ms Under steady-state conditions, the DL-implementation can maintain a sense of DL-time whose maximum long-term mean error relative to the local LAS's sense of DL-time is at most 25 ms for Basic and Link Master DLEs, and 5 ms for Bridge DLEs, under the condition that a single TD DLPDU is received once each 25 s, as shown in Table 71.

NOTE — When connected to a reference node by a multi-link Fieldbus, in which

a) the reference node, two bridges in series, and the node under measurement all meet this requirement; or

b) the reference node and five bridges in series meet the requirements of the 10 ms class, and an additional two bridges in series and the node under measurement all meet the requirement of the 100 ms class,

such a DL-implementation can provide a sense of time differing from that in the reference node by less than 50 ms, and thus from any node similarly removed from the reference node by less than 100 ms.

1 s Under steady-state conditions, the DL-implementation can maintain a sense of DL-time whose maximum long-term mean error relative to the local LAS's sense of DL-time is at most 250 ms for Basic and Link Master DLEs, and 50 ms for Bridge DLEs, under the condition that a single TD DLPDU is received once each 55 s, as shown in Table 71.

NOTE — When connected to a reference node by a multi-link Fieldbus, in which

a) the reference node, two bridges in series, and the node under measurement all meet this requirement; or

b) the reference node and five bridges in series meet the requirements of the 100 ms class, and an additional two bridges in series and the node under measurement all meet the requirement of the 1 s class,

such a DL-implementation can provide a sense of time differing from that in the reference node by less than 0,5 s, and thus from any node similarly removed from the reference node by less than 1 s.

NONE The DL-implementation does not provide a sense of time which is synchronized with that of other nodes on a connected Fieldbus.

NOTES —

1. In this last case, the DL-implementation may still be able to report the last time information heard during a Fieldbus DL-time broadcast, and may even be able to estimate the intervening interval. However, the DL-implementation is unable to maintain a synchronized sense of DL-time with only low-frequency time broadcasts from a reference DLE on the Fieldbus (that is, from the local link's LAS).

2. The time-synchronism class of a link's LAS determines the best achievable time synchronism for the entire link. The time-synchronism class of a bridge determines the best achievable time synchronism for that portion of the extended link which inter-communicates through the bridge.

b) Time-based scheduling services are dependent on the time synchronism of the fieldbus DLE, and so can be provided only when some sense of time synchronism exists within the DLE and on the local link. Are local requests for time-based scheduling services, including consequent scheduling interactions with the local link's LAS DLE, supported?

Table 71 — Maximum permitted phase-tracking error in a DLE's sense of DL-time

at the minimum requireable Time Distribution period

Time-synchronism class	**Maximum permitted error by DLE class**		
	Basic or Link Master DLE	**Bridge DLE**	**value for V(TDP)**
1 µs	250 ns	125 ns	5 ms
10 µs	3 ms	1 ms	50 ms
100 µs	25 ms	10 ms	500 ms
1 ms	250 ms	1/16 ms	5 s
10 ms	2,5 ms	0,5 ms	10 s
100 ms	25 ms	5 ms	25 s
1 s	250 ms	50 ms	55 s
none	> 100 yr	> 100 yr	not applicable

Permitted choices are:

YES PERIODIC schedules, and ONE-TIME schedules with a specified starting time, are supported, with the best achievable time resolution dependent on the time class specified in 11.3(a);

NO these capabilities are not supported.

If the choice specified for category (a) above was NONE, then this value shall be NO.

NOTE — A DLE with no sense of DL-time may still be able to respond to time-based scheduled activities, such as periodic receipt of an ES DLPDU enabling periodic local schedule execution within the DLE. However, such a DLE cannot make a time-based scheduling request to the LAS, because it cannot specify the start time for the request with reference to the LAS' DL-time. Thus all such activities can only occur due to DL-management preconfiguration of the DLE through extra-protocol means, and not as a result of DLS-user requests for DL-service.

c) Time-based scheduling activities within an LAS DLE also are dependent on the time synchronism of the fieldbus DLE, and so can be provided only when some sense of time synchronism exists. Five classes of support for these services have been defined:

DYNAMIC The DL-implementation of LAS functions supports

1) time-based schedules which have been pre-constructed by DL-management;

2) requests for time-based scheduling services received from DLEs on the local link and modification of the current schedule to accommodate those requests, where feasible;

3) reception of the current schedule from a previous LAS DLE; and

4) redistribution of the current schedule to other potential LAS DLEs;

SHARABLE The DL-implementation of LAS functions supports

1) time-based schedules which have been pre-constructed by DL-management;

2) time-based schedules which have been pre-constructed by a previous DYNAMIC-class LAS DLE;

3) reception of the current schedule from a previous LAS DLE; and

4) redistribution of the current schedule to other potential LAS DLEs.

Such a DLE does not have the ability to construct schedules dynamically.

UPDATABLE The DL-implementation of LAS functions supports

1) time-based schedules which have been pre-constructed by DL-management;

2) time-based schedules which have been pre-constructed by a previous DYNAMIC-class LAS DLE; and

3) reception of the current schedule from a previous LAS DLE.

Such a DLE does not have the ability to redistribute the current schedule to other potential LAS DLEs.

NOTE — This inability to redistribute the schedule implies that a received schedule is transformed into an internal representation in which some of the received scheduling information has been discarded.

STATIC The DL-implementation of LAS functions supports time-based schedules which have been pre-constructed by DL-management. It does not have the ability to receive the current schedule from a previous LAS DLE.

NONE Either

1) the DLE is a BASIC DLE; or

2) the implemented LAS functions do not support time-based schedules, but do support free-running schedules which have been pre-constructed by DL-management.

Annex A — (normative) Structure and definition of DL-addresses

(This annex forms an integral part of this standard.)

This clause defines the form of DL-addresses and the usage of various ranges of DL-address components. It includes specific definitions for some standard DL-addresses.

This DL-protocol uses individual (non-group) DL-addresses for other purposes than simply as DLSAP-addresses. The same terminology and the following considerations apply to those non-DLSAP DL-addresses as well.

NOTE — This usage extends the definition of DL-addresses beyond that specified in ISO 7498-3.

A.1 Form of DL-addresses

A standard DL-address may be considered to consist of two parts: link-designator and sub-link selector. The link designator is an unsigned integer, two octets in length. The sub-link selector is also two octets in length.

Link Designator	**Sub-link Selector**
2 octets	2 octets

Figure A.1 — Basic Structure of a DL-address

Most values of the link designator specify an individual link. In these cases, the sub-link selector specifies a unique DL-address within the designated local link, as further described in A.2.2 and A.2.3. When lacking address-specific forwarding information, bridges (relay DLEs) shall base their forwarding decisions only on the link designator portion of such addresses.

However, many values of the link designator specify that the link designator portion of the address is concatenated with the sub-link selector portion to form a non-hierarchical (that is, flat) address. When lacking address-specific information, bridges shall not forward such addresses.

A standard sub-link selector itself consists of two sub-parts: node-designator and sub-node selector. Each of these sub-parts is a small unsigned integer, one octet in length.

implied Link Designator (0000)	**Node Designator**	**Sub-node Selector**
	1 octet	1 octet

Figure A.2 — Basic Structure of a Sub-link Selector

Most values of the node designator specify an individual node. In these cases, the sub-node selector specifies a unique DL-address within the designated local node.

However, some values of the node designator specify that the node designator portion of the address is concatenated with the sub-node selector portion to form a non-hierarchical (that is, flat) sub-address.

This almost-hierarchical structure of link designators, node designators, and sub-node selectors was chosen primarily to facilitate the administration of DL-addresses, and secondarily to reduce the number of entries in bridge address-forwarding tables.

NOTE — Protocol implementations may find that this partially-hierarchical structure is of use during address recognition.

The resulting structural possibilities for DL-addresses are (where "||" indicates concatenation of the nominal link, node and selector designator sub-fields):

Link	**Node**	**Selector**
2 octets	1 octet	1 octet

or

Link	**Node || Selector**
2 octets	2 octets

or

Link || Node || Selector
4 octets

Figure A.3 — DL-address Alternative Structures

A DL-address in which the link designator has a value of zero is known as a *local* address because its context for interpretation is limited to the local link. In contrast, other DL-addresses are known as *extended* addresses, because their interpretation context is the entire extended link.

There are four distinct types of DLS-user-visible DL-address:

a) DLSAP-addresses, which are associated with only one DLSAP in the entire address interpretation context;

b) Group DL-addresses, which are potentially associated with many DLSAPs in one or more DLEs, or with many DLEs;

c) DLCEP-addresses, which are associated

 1) with only one DLCEP and associated DLSAP of a peer-to-peer DLC; or

 2) directly with the publishing DLCEP and associated DLSAP of a multi-peer DLC, and indirectly with all the subscribing DLCEPs and associated DLSAPs of a multi-peer DLC; and

d) DLSEP-addresses, which are associated with one sequence-end-point of a single DLE.

DLSAP-addresses may be used with all of the primitives of the DLS which specify a DLSAP-address. Group DL-addresses shall not be

1) bound to a queue as a source DLSAP-address; or

2) used as a source DLSAP-address in any connection-oriented or connectionless DLS primitive.

Individual non-"DLSAP" DL-addresses may be used as appropriate within the DL-protocol, and where permitted as specific parameters within the DL-services.

A.2 Predefined Values and Ranges for DL-Address Components

Some values of the link-designator or node-designator component of DL-addresses designate a group DL-address. All other combinations of values of the link-designator and node-designator components designate individual DL-addresses. This is summarized in Tables A.1, A.2 and A.3.

NOTE — All link, node and selector values are specified in hexadecimal (radix 16).

A.2.1 Link designators

The link designator component of a DL-address usually designates a specific single local link. Values for the link designator component are predefined, or their usage is constrained, as follows:

0000 — The local link. The zero (0000) value is provided as a convenience for the unification of service primitives, to enable implementations to refer to the local link's address by a constant zero (0000), and to enable isolated-link operation when no bridge is present to identify the link component of an extended link. When the DLPDU-authentication QoS does not specify **MAXIMAL** and the DL-addresses themselves permit it, the link-designator components of all DL-addresses shall be omitted before transmission. Otherwise, the nominal-link-id of the sourcing local link, if known, shall be substituted for the zero (0000) link value before transmission.

0001 … 003F — Non-local group DL-addresses. These link designator values, concatenated with the associated node designator values and selector values — *link* || *node* || *selector* — form an address space of 4 128 768 identifiers usable as group DL-addresses. The resulting DL-addresses do not implicitly designate any specific link.

Non-local group DL-addresses shall be allocated individually or in contiguous blocks. When considered as unsigned 32-bit integers, the first (lowest-valued) 8 such addresses are reserved for use by the DL-protocol; the next 56 such addresses are reserved for standardized group DL-addresses. All of the other addresses are available for use as non-standardized group DL-addresses.

0040 … 007F — Non-local globally-administered group DL-addresses. These link designator values, concatenated with the associated node designator values and selector values — *link* || *node* || *selector* — form an address space of 4 194 304 identifiers for globally-administered group DL-addresses[1]. These DL-addresses are formed by prefixing the 10-bit string "0000 0000 01" to the lower 22 bits of the Organizationally Unique Identifiers specified in the ISO/IEC 8802 family of Local Area Network standards. These DL-addresses do not implicitly designate any specific link.

Table A.1 — link || node || selector addressing

link	N | S	assigned use for specified DL-address range
0000	0000 … FFFF	local link DL-addresses as specified in Table A.2.
0001	0000 … 0007	flat non-local group DL-addresses reserved for DLL use
0001	0008 … 003F	flat non-local group DL-addresses reserved for standard AEs
0001 … 003F	0040 … FFFF	flat non-local group DL-addresses available for vendor, user and dynamic assignment
0040 … 007F	0000 … FFFF	flat non-local globally-administered[1] group DL-addresses — potentially one per vendor
0080	0000 … 0007	flat non-local individual “DLSAP”-addresses reserved for DLL use
0080	0008 … 003F	flat non-local individual DLSAP-addresses reserved for standard AEs
0080 … 00FF	0040 …	flat non-local individual DLSAP-addresses available for vendor, user and dynamic assignment
0100 … 0FFF	0000 … FFBF	flat non-local individual DL-addresses available for vendor, user and dynamic assignment as DLCEP-addresses and DLSEP-addresses
0FFF	FFC0 … FFF7	flat non-local individual DL-addresses reserved for standard AE DLCEP-addresses and DLSEP-addresses
0FFF	FFF8 … FFFF	flat non-local individual DL-addresses reserved for DLL use as DLCEP-addresses and DLSEP-addresses
1000 … FEFF	0000 … FFFF	Individual links of DL-addresses as specified in Table A.2.

[1]These group DL-addresses are based on the Organizationally Unique Identifiers specified in the ISO/IEC 8802 family of Local Area Network standards.

0080 … 00FF	Non-local individual DLSAP-addresses. These link designator values, concatenated with the associated node designator values and selector values — *link* || *node* || *selector* — form an address space of 8 388 608 identifiers usable as individual DLSAP-addresses. The resulting DLSAP-addresses do not implicitly designate any specific link or node. Non-local individual DLSAP-addresses shall be allocated individually or in contiguous blocks. When considered as unsigned 32-bit integers, the first (lowest-valued) 8 such addresses are reserved for use by the DL-protocol as “DLSAP”-addresses; the next 56 such addresses are reserved for standardized individual DLSAP addresses for standard AEs. All of the other addresses are available for use as non-standardized individual DLSAP addresses.
0100 … 0FFF	Non-local individual non-DLSAP DL-addresses usable as DLCEP-addresses and DLSEP-addresses. These link designator values, concatenated with the associated node designator values and selector values — *link* || *node* || *selector* — form an address space of 251 658 240 identifiers usable as individual DLCEP- and DLSEP-addresses. The resulting DL-addresses do not implicitly designate any specific link or node. Non-local individual non-DLSAP DL-addresses shall be allocated individually or in contiguous blocks. When considered as unsigned 32-bit integers, the last (highest-valued) 8 such addresses are reserved for use by the DL-protocol; the next last 56 such addresses are reserved for standardized individual DLCEP-addresses or DLSEP-addresses for standard AEs. All of the other addresses are available for use as non-standardized individual DLCEP addresses and DLSEP-addresses.
1000 … FEFF	Designators of individual links. Link designators shall be allocated, individually or in contiguous blocks, in order of increasing address values when considered as unsigned 16-bit integers. The value of the DL-management configuration parameter V(ML) shall be greater or equal to the largest of the link-designator values assigned to links of the extended link.

NOTE — This parameter is used by bridges to assist in the provision of default forwarding actions for each link.

NOTE 1 — The potential link-designator values FF00 – FFFF are reserved to support the implementation of identifier-references to DL-addresses as noted in (see 12.8.2.4, 17.2.2.1, 17.2.2.2, 17.2.2.3, 24.1.2.1.1, 24.1.2.1.2, 24.2.2.1 of) Part 3 of this International Standard.

In summary, each DLE capable of recognizing LONG DL-addresses (see 6.2.2) recognizes link designator values 0001 – 0FFF as designating the extended group and individual address spaces — *link* || *node* || *selector* — common to the extended link.

Each DLE capable of recognizing LONG DL-addresses recognizes link designator value 0000 as designating the nominal-link, which is a single local link designator value ≥ 1000. The resulting identifier values — *nominal-link-id* || *node* || *selector* — form an address space of up to 65 280 identifiers usable as individual and group DL-addresses as described in A.2.2.

NOTE 2 — The *link* || *node* || *selector* components of these address identifiers take the hexadecimal form WWWWXXYY, where 1000 ≤ WWWW, 01 ≤ XX ≤ FF and 00 ≤ YY ≤ FF. When 01 ≤ XX ≤ 03, such addresses are group DL-addresses; all other such addresses are individual DL-addresses. Addresses where 1000 ≤ WWWW and XX = 00 refer to the local node, and are valid only within the software of that node.

When the address space needs of a given link exceed that provided by a single nominal-link-id, secondary link-ids may be allocated to that link. For each such secondary link-id, the resulting values — *secondary-*

link-id || *node* || *selector* — form an additional address space of 57 600 identifiers usable as additional individual and group DL-addresses for the link as described in A.2.2.

NOTE 3 — The *link* || *node* || *selector* components of these additional address identifiers take the hexadecimal form WWWWXXYY, where 1000 ≤ WWWW, and either 0140 ≤ XXYY ≤ 03FF, or 0440 ≤ XXYY ≤ 0FBF, or 10 ≤ XX ≤ FF and 08 ≤ YY ≤ F7. When 01 ≤ XX ≤ 03, such address are group DL-addresses; all other such addresses are individual DL-addresses.

DLEs which are not capable of recognizing LONG DL-addresses recognize only SHORT and VERY-SHORT addresses (see 6.2.2), which always designate the nominal-link. Such DLEs are not capable of directly communicating or being directly managed through a bridge.

A.2.2 Node designators

The node designator component of a DL-address usually designates a single DLE. When the link designator component does not have the value 0001 – 0FFF, then the values for the node designator component are predefined, or their usage is constrained, as follows:

Table A.2 — link-local node || selector addressing

N	S	assigned use for specified DL-address range
00	00 … FF	local node DL-addresses as specified in Table A.4.
01	00 … 07	flat link-local group DL-addresses reserved for DLL use
01	08 … 3F	flat link-local group DL-addresses reserved for standard AEs
01 … 03	40 … FF	flat link-local group DL-addresses available for vendor, user and dynamic assignment
04	00 … 07	flat link-local individual “DLSAP”-addresses reserved for DLL use
04	08 … 3F	flat link-local individual DLSAP-addresses reserved for standard AEs
04	40 … FF	flat link-local individual DLSAP-addresses available for vendor, user and dynamic assignment
05 … 0F	0 … BF	flat link-local individual DL-addresses available for vendor, user and dynamic assignment as DLCEP-addresses and DLSEP-addresses
0F	C0 … F7	flat link-local individual DL-addresses reserved for standard AE DLCEP-addresses and DLSEP-addresses
0F	F8 … FF	flat link-local individual DL-addresses reserved for DLL use as DLCEP-addresses and DLSEP-addresses
10 … FF	00 … FF	Individual nodes of DL-addresses as specified in Table A.4

00	The local DLE. Equivalent to a DLE-internal address. This designator is provided as a convenience for the unification of service primitives, and to enable implementations to refer to the local DLE address by a constant zero (00). When transmission is appropriate, an implementation shall substitute the DLE's own nominal-node-id for this zero (00) address. NOTE 1 — As a consequence, DLPDUs with a node designator component of zero (00) in a DL-address (other than a flat DL-address) are never transmitted, and shall be discarded upon reception. This node designator shall not be used with secondary-link-ids.
01 … 03	Link-local group DL-addresses. These node designator values, concatenated with the nominal or a secondary link-id of the local link and the associated selector values — *link-id* || *node* || *selector* — form address spaces of 768 identifiers each, usable as group DL-addresses. Link-local group DL-addresses shall be allocated, individually or in contiguous blocks, in order of increasing address values when considered as unsigned 16-bit integers. The first (lowest-valued) 8 such addresses are reserved for use by the DL-protocol; the next 56 such addresses are reserved for standardized group DL-addresses. All of the other addresses are available for use as non-standardized group DL-addresses
04	Link-local individual DLSAP-addresses. These node designator values, concatenated with the nominal or a secondary link-id of the local link and the associated selector values — *link-id* || *node* || *selector* — form address spaces of 256 identifiers each, usable as individual DLSAP-addresses. The resulting DLSAP-addresses do not implicitly designate any specific node. Link-local individual DLSAP-addresses shall be allocated, individually or in contiguous blocks, in order of increasing address values when considered as unsigned 16-bit integers. The first (lowest-valued) 8 such addresses are reserved for use by the DL-protocol as "DLSAP"-addresses; the next 56 such addresses are reserved for standardized individual DLSAP addresses for standard AEs.
05 … 0F	Link-local individual non-DLSAP DL-addresses usable as DLCEP-addresses and DLSEP-addresses. These node designator values, concatenated with the nominal or a secondary link-id of the local link and the associated selector values — *link-id* || *node* || *selector* — form address spaces of 2 816 identifiers usable as individual DLCEP- and DLSEP-addresses. The resulting DL-addresses do not implicitly designate any specific node. Link-local individual non-DLSAP DL-addresses usable with connected DL-services shall be allocated, individually or in contiguous blocks, in order of decreasing address values when considered as unsigned 16-bit integers. The first (highest-valued) 8 such addresses are reserved for use by the DL-protocol; the next 56 such addresses are reserved for standardized individual DLCEP-addresses or DLSEP-addresses for standard AEs. All of the other addresses are available for use as non-standardized individual DLCEP addresses and DLSEP-addresses.
10 … FF	Designators of individual nodes. These node designator values, concatenated with the nominal link-id of the local link, provide the equivalent of a "unique hardware address" for each individual physical DLE. Node designators shall be allocated, individually or in contiguous blocks, as shown in table A.3:

Table A.3 — link-local node designators

N	assigned use for specified node designator range
00	local DLE
01 … 0F	unusable
10 … 13	bridge-class DLEs
14 …	link-master-class DLEs
…	unused
… F7	basic-class DLEs
F8 … FB	non-"visitor" DLEs awaiting proper node designator assignment
FC … FF	"visitor" DLEs

00 The local DLE. Equivalent to a DLE-internal address. This designator is provided as a convenience for the unification of service primitives, and to enable implementations to refer to the local DLE address by a constant zero (00). When transmission is appropriate, an implementation shall substitute the DLE's own nominal-node-id for this zero (00) address.

NOTE 2 — As a consequence, DLPDUs with a node designator component of zero (00) in a DL-address (other than a flat DL-address) are never transmitted, and shall be discarded upon reception.

01 … 0F Reserved for unrelated use within the DL-protocol – not available for DLE NODE DL-addresses.

10 … 13 Reserved for allocation to DLEs which bridge (interconnect) the local link (whose link-id is the nominal link-id) to other links, and in particular to higher-capacity links.

NOTE 3 — Such bridges are expected to serve as link-local managers for the DL-protocol.

14 … Reserved for allocation to other DLEs which are capable of initializing and managing the local link, and which are intended to be permanently attached to the local link (not, for example, maintenance DLEs such as "hand-held communicators").

These node designators shall be allocated, individually or in contiguous blocks, in order of increasing address values when considered as unsigned 8-bit integers.

… Reserved for allocation to DLEs which require more local DL-addresses than are provided hierarchically by the DLE's primary node designator.

… F7	Reserved for allocation to DLEs which are not capable (or are configured to be incapable) of initializing the local link. These node designators shall be allocated, individually or in contiguous blocks, in order of decreasing address values when considered as unsigned 8-bit integers.
F8 … FB	Reserved for temporary use by other DLEs which are intended to be permanently attached to the local link while they determine their appropriate final node designator.
FC … FF	Reserved for allocation to DLEs which are not intended to be permanently attached to the local link (for example, maintenance DLEs such as “hand-held communicators). NOTE 4 — Such “visitor” DLEs normally are capable of initializing and managing the local link, but are the least desirable such DLEs for this purpose since their long-term presence on the link is not expected.

When not used to designate a single DLE, node designator values between those used for DLEs capable of initializing the local link, and those which are incapable (or are configured to be incapable) of initializing the local link, can be concatenated with the nominal or a secondary link-id of the local link, and with selector components whose value is ≥ 08 and ≤ F7, to form additional identifier spaces of 240 individual DL-addresses each — *link-id* || *node* || *selector*. Each such additional address space may be assigned to a single node, or may be shared among the nodes of the link.

NOTE 5 — The *node* || *selector* components of these additional address identifiers take the hexadecimal form XXYY, where 10 < XX < FC and 08 ≤ YY ≤ F7.

In summary, when the link designator value designates the local link, then each DLE recognizes node designator values 01 – 0F as designating link-local group and individual address spaces — *link-id* || *node* || *selector* — common to all nodes on the local link.

Each DLE recognizes node designator value 00 as designating a single local node designator value ≥ 10 (hexadecimal), the nominal-node. The resulting identifier values — *nominal-link-id* || *nominal-node-id* || *selector* — form a space of identifiers usable as individual DL-addresses as described in A.2.3.

When the address space needs of a given node exceed that provided by a single nominal-node-id, secondary node-ids may be allocated to that node. For each such secondary node id, the resulting values — *link-id* || *secondary-node-id* || *selector* — with selector component values ≥ 08 and ≤ F7, form an additional space of 240 identifiers usable as additional individual DL-addresses for the node as described in A.2.3. The same method may be used to extend the set of non-node specific individual addresses for the link.

A.2.3 Selectors

The selector component of a DL-address usually designates a single protocol entity within the DLE.

When the link designator component does not have the value 0001 – 0FFF, and when the node designator component does not have the value 01 – 0F, then the values for the selector component are predefined, and their usage is constrained, as follows:

00	The local DLE. Equivalent to a DLE-internal address. This designator is provided as a convenience for the unification of service primitives, and to enable implementations to refer to the local DLE address by a constant zero (00). When transmission is appropriate, an implementation shall substitute the DLE’s own nominal-node-id for this zero (00) address. NOTE 2 — As a consequence, DLPDUs with a node designator component of zero (00) in a DL-address (other than a flat DL-address) are never transmitted, and shall be discarded upon reception.

Table A.4 — node-local selector addressing

selector	assigned use for specified DL-address range
00 … 01	node-local individual “DLSAP”-addresses reserved for DLL use
02 … 07	node-local individual DLSAP-addresses reserved for standard AEs
08 … 1F	node-local individual DLSAP-addresses available for vendor, user and dynamic assignment
20 … F7	node-local individual DL-addresses available for vendor, user and dynamic assignment as DLCEP-addresses and DLSEP-addresses
F8 … FF	node-local individual DL-addresses reserved for use as standard AE DLCEP-addresses and DLSEP-addresses

01 … 0F	Reserved for unrelated use within the DL-protocol - not available for DLE NODE DL-addresses.
10 … 13	Reserved for allocation to DLEs which bridge (interconnect) the local link (whose link-id is the nominal link-id) to other links, and in particular to higher-capacity links. NOTE 3 — Such bridges are expected to serve as link-local managers for the DL-protocol.
14 …	Reserved for allocation to other DLEs which are capable of initializing and managing the local link, and which are intended to be permanently attached to the local link (not, for example, maintenance DLEs such as “hand-held communicators”). These node designators shall be allocated, individually or in contiguous blocks, in order of increasing address values when considered as unsigned 8-bit integers.
…	Reserved for allocation to DLEs which require more local DL-addresses than are provided hierarchically by the DLE’s primary node designator.
… F7	Reserved for allocation to DLEs which are not capable (or are configured to be incapable) of initializing the local link. These node designators shall be allocated, individually or in contiguous blocks, in order of decreasing address values when considered as unsigned 8-bit integers.
F8 … FB	Reserved for temporary use by other DLEs which are intended to be permanently attached to the local link while they determine their appropriate final node designator.

FC … FF	Reserved for allocation to DLEs which are not intended to be permanently attached to the local link (for example, maintenance DLEs such as “hand-held communicators"). NOTE 4 — Such “visitor” DLEs normally are capable of initializing and managing the local link, but are the least desirable such DLEs for this purpose since their long-term presence on the link is not expected.

A.3 Predefined DL-Addresses

As indicated in Tables A.1 through A.3, some specific DL-addresses are defined in this Annex. These DL-addresses are defined to facilitate provision of DL-services and initialization of OSI communications within a physically distributed real system.

A.3.1 Predefined flat non-local DL-addresses

A number of flat non-local DL-addresses are defined within this Annex, as specified in Table A.5.

NOTE — SLAE is the System Load Application Entity; SMAE is the System Management Application Entity.

Table A.5 — predefined flat non-local DL-addresses

link	**|| N || S**	
0001	0000	the DL-support functions of “all” (see note 1) DLEs on the extended link
0001	0001	the DL-support functions of “all” (see note 1) LM DLEs on the extended link
0001	0002	the DL-support functions of “all” (see note 1) bridge DLEs on the extended link
0001	0003	the DL-bridge functions of “all” (see note 1) bridge DLEs on the extended link
0001	0008	the SMAEs of “all” (see note 1) unconfigured DLEs on the extended link
0001	0009	the SMAEs of “all” (see note 1) DLEs on the extended link
0001	000A	the SMAEs of “all” (see note 1) LM DLEs on the extended link
0001	000B	the SMAEs of “all” (see note 1) bridge DLEs on the extended link
0001	000C	the SLAEs of “all” (see note 1) LoadServers on the extended link
0001	000D	the SLAEs of “all” (see note 1) LoadableDevices on the extended link
0080	0000	reserved for DLL use for a DL-support “DLSAP”-address (see note 1)
0080	0004	the “DLSAP”-address for the DL-bridge functions of the bridge DLE on the extended link which is the root bridge of the spanning tree
0FFF	FFFE	the DLCEP-address for the DLC from the bridge functions of the bridge DLE on the extended link which is the root bridge of the spanning tree to the bridge functions in all other bridge DLEs on the extended link
0FFF	FFFF	reserved for DLL use for a DL-support DLCEP-address or DLSEP-address (see note 1)
NOTE — DLEs which do not recognize LONG DL-addresses are necessarily excluded from these sets.		

A.3.2 Predefined flat local DL-addresses

A number of flat local DL-addresses are defined within this Annex, as specified in Table A.6. These correspond one-for-one with the predefined flat non-local DL-addresses specified in Table A.5. The extent of the addresses in Table A.6 is the local link; the extent of the addresses in table A.5 is the entire extended link.

The DL-addresses in Table A.6 have a one-for-one correspondence with analogous DL-addresses in Table A.5; while the addresses in table A.6 refer to functions local to each link, the analogous addresses of Table A.5 refer to similar functions global to the entire extended link. This addressing correspondence is intentional, and may serve to ease implementation complexity. Thus this correspondence should be preserved in future address assignments.

Table A.6 — predefined flat link-local DL-addresses

node || selector		
01	00	the DL-support functions of all DLEs on the link
01	01	the DL-support functions of all LM DLEs on the link
01	02	the DL-support functions of all bridge DLEs on the link
01	03	the DL-bridge functions of all bridge DLEs on the link
01	08	the SMAEs of all unconfigured DLEs on the link
01	09	the SMAEs of all DLEs on the link
01	0A	the SMAEs of all LM DLEs on the link
01	0B	the SMAEs of all bridge DLEs on the link
01	0C	the SLAEs of all LoadServers on the link
01	0D	the SLAEs of all LoadableDevices on the link
04	00	the “DLSAP”-address for the DL-support functions of the DLE on the link which is serving as LAS
04	04	the “DLSAP”-address for the DL-bridge functions of the bridge DLE on the link which is dominant (closest to the root) in the bridge spanning tree
0F	FE	the DLCEP-address for the DLC from the bridge functions of the bridge DLE on the link which is dominant (closest to the root) in the bridge spanning tree to the bridge functions in all other bridge DLEs on the link
0F	FF	the DLCEP-address for the DLC from the DL-support functions of the DLE on the link which is serving as LAS to the DL-support functions of all of the other LM DLEs on the link

A.3.3 Predefined node-local DL-addresses

A number of node-local DL-addresses are defined within this Annex, as specified in Table A.7.

Table A.7 — predefined node-local DL-addresses

selector	**assigned use for specified DL-address**
00	the “DLSAP”-address for the DL-support functions of the node’s DLE
01	the “DLSAP”-address for the DL-bridge functions of the node’s DLE
02	the DLSAP-address for the node’s SMAE
03	the DLSAP-address for the node’s SLAE

A.4 Representation of DL-Addresses as locally-administered 48-bit MAC-addresses

The ISO/IEC Local Area Network protocols [ISO/IEC 8802] provide a foundation for portions of the IEC Fieldbus protocols, such as the Ph-redundancy state machine, the inter-DL-relay-(bridge)-protocol, and the AL-system-load-protocol. The latter two protocols, which specify six-octet encodings for each ISO/IEC MAC-address, are being adapted for Fieldbus use with the minimum of necessary changes; thus there is a need to specify the representation of Fieldbus four-octet DL-addresses for use with the Fieldbus adaptations of these protocols.

A six-octet ISO/IEC MAC-address, as specified in ISO/IEC 18802 (ANSI/IEEE Std 802), contains, in order of transmission

a) an initial bit, I/G, indicating whether the MAC-address is an individual MAC-SAP-address (I/G=0) designating a single MAC-sublayer-entity, or a group MAC-address (I/G=1) designating a group of zero or more MAC-sublayer-entities;

b) a second bit, U/L, indicating whether the MAC-address contains a globally-administered component (U/L=0), or is completely locally-administered (U/L=1); and

c) a 22-bit globally-administered component and a 24-bit vendor-administered component when U/L=0, or a 46-bit user-administered component when U/L=1.

Universally-administered MAC-addresses			
I/G	U/L	globally-administered	vendor-administered
1 b	1 b	22 bits	24 bits

Locally-administered MAC-addresses		
I/G	U/L	user-administered
1 b	1 b	46 bits

Figure A.4 — Basic Structure of MAC-addresses

Since DL-addresses are not administered so as to be globally-unique they shall be represented as six-octet locally-administered MAC-addresses as shown in Figure A.5.

a DL-address as a locally-administered MAC-address			
I/G	U/L	set by DL-management	extended DL-address
1 b	1 b	14 bits	32 bits

Figure A.5 — Representation of a DL-address as a MAC-address

The 14-bit MAC-address component set by DL-management shall be set equal to the value of the extended-link's **DL-MAC-address-embedding-prefix** parameter, V(MEP), which shall be distributed as part of the extended-link's configuration information. The default value for this parameter shall be hexadecimal 3FFF (all ones), but it shall be changeable by DL-management to permit more than one extended link to be directly addressable from inter-working ISO/IEC local area networks.

Each 32-bit DLSAP-, DLCEP- and DLSEP-address shall be represented as a six-octet MAC-address by prefixing the 32-bit non-group DL-address with the binary string "01" followed by the extended-link's 14-bit DL-MAC-address-embedding-prefix, V(MEP).

Each 32-bit group DL-address shall be represented as a six-octet MAC-address by prefixing the 32-bit group DL-address with the binary string "11" followed by the extended-link's 14-bit DL-MAC-address-embedding-prefix, V(MEP).

In addition, should there be a need to specify a six-octet MAC-address for the group of multi-peer subscriber DLCEPs associated with a specified multi-peer publisher DLCEP, then the implicit address of that group of subscriber DLCEPs shall be represented as a six-octet MAC-address by prefixing the publisher's 32-bit individual DLCEP-address with the binary string "11" followed by the extended-link's 14-bit DL-MAC-address-embedding-prefix, V(MEP).

Annex B — (normative) DL-support sub-protocol

(This annex forms an integral part of this standard.)

B.1 Scope

This annex to this part of this International Standard

a) defines the encodings for the higher-level (see 5.1.3) support PDUs (SPDUs) required to support the LAS operation and scheduling sub-protocol;

b) defines the components of a defined sequence (see 29.3.2.1) of Part 3 of this International Standard; and

c) defines the encodings for the higher-level (see 5.1.3) support PDUs (SPDUs) required to support various configuration and capability queries used by

— the DL-SUBSCRIBER-QUERY request and DL-LISTENER-QUERY request elements of procedure;

— bridges for forwarding-database maintenance; and

— specialized DLL monitoring, analysis and management tools.

B.2 Overview of LAS operation

An LAS is required to perform the following functions:

a) Link maintenance — the LAS

1) detects the presence of a newly added DLE on the local link, after which it activates that DLE;

2) detects that a previously participating DLE is no longer active on the local link;

3) maintains a list of all DLEs operational on the local link; and

4) distributes the list of operational DLEs to all other link-master DLEs on the local link.

b) Time distribution — the LAS acts as the time master for the local link and sends time distribution DLPDUs to all other DLEs on the local link to provide a common sense of DL-time for the local link.

c) Token circulation — the LAS sends a PT DLPDU to each active DLE on the local link. By doing this repeatedly, the LAS DLE provides a delegated token which “circulates” successively, usually in NODE DL-address order, to all active DLEs on the local link.

d) Schedule execution — the LAS sends DLPDUs on the local link as dictated by a schedule which specifies the initial DL-time and repetition period for each of those transmissions. An initial schedule (possibly null) is provided by DL-management, and may be subsequently modified by the active LAS.

e) LAS transfer — the LAS transfers its role to another link master DLE, if requested by that link master or by DL-management.

An LAS may also perform the following functions:

f) Schedule construction — the LAS may process scheduling requests from other DLEs on the local link and modify the existing schedule, if any, to satisfy those requests. The schedule construction is required to take into account the maximum-scheduled-traffic, V(MST), (see 5.7.5.5) and maximum-scheduling-overhead, V(MSO), (see 5.7.5.6).

g) Schedule transfer — the LAS may send the current state of the link's schedule to another DLE on the local link, either as a backup measure or preceding an intentional transfer of the LAS role. This annex specifies the SPDUs required to transfer the pre-constructed schedule from the LAS DLE to another LM DLE on the local link. The format of these SPDUs does not support the updating of such a transferred schedule. The procedures for schedule transfer are specified in 10.2.10, 10.2.11, 10.3.7 and 10.3.8.

The procedures for LAS operation are specified in 10.2; procedures for DL-support operation are specified in 10.3; procedures for schedule construction requests are specified in 9.4.3; procedures for time distribution are specified in 9.4.1; procedures for token circulation are specified in 7.15; procedures for local schedule construction requests are specified in 9.4.3. The SPDUs for transferring the schedule construction requests to the LAS DLE are specified in this annex. The procedures for schedule construction depend upon the implementation and thus are not specified.

This annex also specifies SPDUs to support bridges, protocol-analyzers and other diagnostic tools, (see B.3.2.7, B.3.2.8, B.3.2.9, B.3.2.10, B.3.6.4, and B.3.6.5). The procedures for sending and receiving these SPDUs depend upon the implementation and thus are not specified.

B.3 DL-support subprotocol definition

The DL-support subprotocol defines Support Protocol Data Unit (SPDU) encodings to support the needs of LAS operation, including scheduling and other DLE functions. Other DLE support functions presently defined include mechanisms to share address and limited configuration information in support of

— DL-SUBSCRIBER-QUERY (see 9.2.3) and DL-LISTENER-QUERY (see 9.3.4) requests;

— filtering-database-maintenance within bridges (see Annex C/C.4(9)); and

— network analysis (link analyzer) support tools.

Any DLPDU sent to, or by, the DL-support functions within a DLE, including any DLPDU addressed to a NODE DL-address, which has a non-null user-data field, shall contain as "user data" a single SPDU whose encoding and interpretation is as described in this sub-clause. This requirement includes any DLPDU addressed to a DLSAP-address designating LAS functionality, such as link-local DL-address 0400_{16}. It also includes any PR or TL DLPDU, both of which always have a user-data field.

B.3.1 Common definitions

B.3.1.1 SPDU header

The first octet of each SPDU shall specify a header common to all SPDUs, as depicted in Table B.1. It shall contain:

a) an SPDU classification; and

b) for some SPDUs, the protocol version, whose value is specified in 8.

Table B.1 — SPDU 1st octet: SPDU class, and protocol version or subclass

SPDU class					protocol version or subclass		
CCCCC $\leq$ 11011					VVV		
CCCCC $\geq$ 11100					SSS		
7	6	5	4	3	2	1	0

When the SPDU class is in the range 00 to $1B_{16}$, then the protocol version, VVV, whose value is specified in 8, shall be present and shall be encoded as in 8.1(a.4).

NOTE 1 — The resulting range of values for the octet is 00 to DF_{16}.

When the SPDU class is in the range $1C_{16}$ to $1F_{16}$, then the subclass field, SSS, shall be present.

NOTE 2 — The resulting range of values for the octet is $E0_{16}$ to FF_{16}.

Some of the SPDUs contain only the header octet. The SPDUs fall into five classes:

a) SPDUs supporting normal link initialization, DLE activation, and other link maintenance functions;

b) SPDUs supporting LAS transfer;

c) SPDUs supporting schedule construction;

d) SPDUs supporting schedule transfer; and

e) SPDUs supporting non-LAS features of the DL-protocol, such as DL-SUBSCRIBER-QUERY and DL-LISTENER-QUERY requests, bridges, remote DL-management, and special-purpose DL-protocol analyzers.

B.3.1.2 Booleans

All Boolean elements share a common encoding:

0 FALSE; and

1 TRUE.

B.3.2 Link-maintenance SPDUs

B.3.2.1 Probe-response SPDU

The probe-response SPDU is sent in a Probe Response (PR) DLPDU (see 7.14.1). The probe-response SPDU shall be formatted as specified in Table B.2. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.2 — Probe-response SPDU

octet index	contents of field							
1	0000 0VVV							
2	DL-protocol versions supported							
3	N	F	00					
4	V(MICD)							
5 **6**	V(MRD) × V(ST)							
7	V(MID)							
8	V(RID)							
	7	6	5	4	3	2	1	0

a) The SPDU header octet, octet 1, shall be encoded as 0000 0VVV, which identifies this as a DL-probe-response SPDU.

Table B.3 — DL-protocol versions supported

DL-protocol versions supported							
V7	V6	V5	V4	V3	V2	V1	V0
7	6	5	4	3	2	1	0

b) Octet 2 shall specify the version(s) of the DL-protocol supported, as depicted in Table B.3, where each V_K shall be encoded as a Boolean indicating whether protocol version K is supported.

Table B.4 — PR-SPDU: 3rd and 4th octets

token circulation not needed	**fractional duty cycle operation**	**reserved**		**maximum-inactivity-to-claim-LAS-delay (msbs)**			
N	F	00		V(MICD)/256			
15	14	13	12	11	10	9	8

maximum-inactivity-to-claim-LAS-delay (LSB)							
V(MICD) modulo 256							
7	6	5	4	3	2	1	0

c) Octets 3 and 4 shall specify, as depicted in Table B.4:

1) the DLE's lack of need for token circulation without an explicit request, encoded as a Boolean, N: 0 (no, token circulation is needed) or 1 (yes, token circulation is not needed);

2) whether the DLE will function as an FDC DLE which can be expected to be non-responsive to some live-list link-maintenance queries, and whether that DLE should be included in the expected-non-response list, V(ENRL) (see 5.7.5.4), encoded as a Boolean, F;

3) reserved for future use, encoded as 00; and

4) if the responding DLE is a link master then that DLE's required maximum-inactivity-to-claim-LAS-delay, V(MICD) defined in 5.7.5.19, else zero.

NOTE — If this value is larger than the link's configured value of V(MICD), then the responding DLE is not capable of operating as LAS on the link as specified in 10.3.5.

d) Octets 5 and 6 shall specify the responding DLE's required value of maximum-response-delay-in-octets, V(MRD) × V(ST), defined in 5.7.1.3 and 5.7.1.1. If this value is larger than the link's configured value of maximum-response-delay-in-octets, V(MRD) × V(ST), then the responding node is not capable of operating on the local link.

e) Octet 7 shall specify the responding DLE's required value of minimum-inter-DLPDU-delay, V(MID) defined in 5.7.1.12. If this value is larger than the link's configured value of V(MID), then the responding DLE is not capable of operating on the local link.

f) Octet 8 shall specify the value for the sending DLE's random identifier variable, V(RID) (see 5.7.1.18), which is rerandomized upon receipt of a PN DLPDU addressed to the DLE.

B.3.2.2 Node-activation SPDU

The node-activation SPDU specifies the DL-configuration parameters for the receiving DLE's link. Its primary purpose is to permit the early initiation of DL-services and thus permit higher-layer protocol stacks, including the OSI management protocol stack specified in Part 7 of this International Standard, to become fully operational.

The node-activation SPDU shall be formatted as specified in Table B.5. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.5 — Node-activation SPDU

octet index	contents of field	
1	0000 1VVV	
2	value for V(MRC)	value for V(NRC)
3	value for V(PhLO)	
4	reserved	value for V(TSC)
5 **6**	value for V(MEP)	
7 **8**	value for V(TL)	
9 **10**	value for V(NDL)	
11	V(RID)	
	7 6 5 4 3 2 1 0	

a) The SPDU header octet, octet 1, shall be encoded as 0000 1VVV, which identifies this as a node-activation SPDU**.**

b) Octet 2 shall specify in two quartets the value for the receiving DLE's maximum-retry-count variable, V(MRC), as specified in 5.7.1.5, and the value for the receiving DLE's network-retry-count variable, V(NRC), as specified in 5.7.1.6. Both values are properties of the local link.

c) Octet 3 shall specify the value for the per-DLPDU-PhL-overhead, as specified in 5.7.1.2. When the value for this variable is not configured, a value corresponding to the LAS DLE's own associated PhE, as then configured, shall be used.

Table B.6 — Node-activation SPDU: 4th octet

reserved					V(TSC)		
7	6	5	4	3	2	1	0

d) Octet 4 shall specify, as depicted in Table B.6:

1) reserved for future use, encoded as 00000; and

2) value to be used by the receiving DLE as the local link's time-synchronization-class, V(TSC), as specified in 5.7.1.25.

e) Octets 5 and 6 shall specify the value for the receiving DLE's DL-MAC-address-embedding-prefix variable, V(MEP), as specified in 5.7.1.10.

f) Octets 7 and 8 shall specify the value for the receiving DLE's this-link variable, V(TL), as specified in 5.7.1.9. When the appropriate value for this variable is unknown, the LAS DLE shall use a value of zero.

g) Octets 9 and 10 shall specify the value for the receiving DLE's network-DLPDU-lifetime variable, V(NDL), as specified in 5.7.1.7.

h) Octet 11 shall specify the value of the addressed DLE's random identifier variable, V(RID) (see 5.7.1.18), as recorded by the LAS from a prior-received probe-response SPDU.

B.3.2.3 LAS-data-base-status SPDU

The LAS-data-base-status SPDU is sent by the LAS to all LM DLEs on the local link to publish the latest version number for the active schedule and the latest revision number for the local-link live-list, V(LL) (see 5.7.5.2). It is also sent in each TL DLPDU. The LAS-data-base-status SPDU shall be encoded as specified in Table B.7. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.7 — LAS-data-base-status SPDU

octet index	contents of field
1	1111 0000
2	schedule-type
3 4	schedule version-number
5 6	live-list revision-number

a) The SPDU header octet, octet 1, shall be encoded as 1111 0000, specifying an LAS-data-base-status SPDU.

Table B.8 — LAS-data-base-status SPDU: 2nd octet

schedule-type							
T	reserved					D	S
7	6	5	4	3	2	1	0

b) Octet 2 shall specify, as depicted in Table B.8:

1) the LAS's capability to transfer its schedule, T, encoded as a Boolean: 0 (no, LAS is not capable) or 1 (yes, LAS is capable of transferring its schedule to other LM DLEs);

2) reserved for future use, encoded as 00000;

3) whether all or part of the active schedule has been dynamically constructed by the LAS, D, encoded as a Boolean; and

4) whether all or part of the active schedule has been statically constructed by DL-management, S, encoded as a Boolean;

NOTE — if the LAS has an active schedule, the two subfields D and S both cannot be encoded as 0.

c) Octets 3 and 4 shall specify the version-number of the currently active schedule. If there is no active schedule, then its value shall be zero; otherwise it shall have a non-zero value.

d) Octets 5 and 6 shall specify the current live-list revision-number. It is always a non-zero value, which is incremented (modulo 2^{16}) every time the LAS detects a change in the live-list.

B.3.2.4 Live-list-change SPDU

The Live-list-change SPDU is sent by the LAS to all LM DLEs on the local link, whenever the LAS has detected a change in the status of one or more DLEs. This SPDU shall be encoded as specified in Table B.9. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.9 — Live-list - change SPDU

octet index	contents of field
1	1111 0001
2 3	live-list revision-number
4 5	DLE-status
…	DLE-status

a) The SPDU header octet, octet 1, shall be encoded as 1111 0001, specifying an Live-list-change SPDU.

b) Octets 2 and 3 shall specify the live-list revision-number. It is incremented every time the LAS detects a change in the live-list and prepares the live-list-SPDU, but it is not incremented if the change is added to an SPDU which has not yet been published by the LAS. The value in this SPDU is the revision-number which applies to the live-list immediately after the changes specified in this SPDU. When this number reaches its maximum value, the next time it is incremented, it is incremented to 1 (thus this number is always non-zero).

c) The remainder of the SPDU is an array of two octet members specifying DLE-status and structured as shown in Table B.10:

1) The first octet of each member shall specify the NODE DL-address of the DLE which is affected by the change.

2) The second octet of each member shall specify the status of the DLE, encoded as:

i) N, the DLE's non-need for token circulation, as returned by that DLE in a previously sent PR SPDU (see B.3.2.1(c.1));

ii) F, whether the DLE is an FDC DLE, as returned by that DLE in a previously sent PR SPDU (see B.3.2.1(c.2));

iii) reserved for future use, encoded as 0000; and

iv) SS, the last-observed status of that DLE, encoded as

01: not present;

10: present but presumed now asleep, which is only possible for FDC DLEs; or

11: present and awake.

Table B.10 — DLE-status structure

octet sub-index	**contents of subfield**							
1	NODE DL-address of the described DLE							
2	N	F	reserved				SS	
	7	6	5	4	3	2	1	0

The length of this SPDU may approach 128 octets.

B.3.2.5 Live-list-request SPDU

This SPDU requests that the current LAS DLE publish the entire live-list. The Live-list-request SPDU shall be encoded with the format specified in Table B.1, as follows:

a) The SPDU header octet, octet 1, shall be encoded as 1111 0010, specifying a Live-list -request SPDU.

B.3.2.6 Live-list-detail SPDU

The Live-list-detail SPDU is sent by the LAS to all LM DLEs on the local link. This SPDU shall be encoded as specified in Table B.11. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.11 — Live-list-detail SPDU

octet index	contents of field
1	1111 0011
2	NODE DL-address of LAS's LM DLE
3 **4**	live-list revision-number
5	V(FUN)
6	V(NUN)
7 **…**	Liveness-status
… **…**	DLE-type

a) The SPDU header octet, octet 1, shall be encoded as 1111 0011, specifying a Live-list-detail SPDU.

b) Octet 2 shall specify the NODE DL-address of the link-master DLE which is currently providing the LAS functionality.

c) Octets 3 and 4 shall specify the current live-list revision-number.

d) Octet 5 shall specify the current value of first-unpolled-node, V(FUN), (see 5.7.5.15).

e) Octet 6 shall specify the current value of number-of-consecutive-unpolled-nodes, V(NUN), (see 5.7.5.16).

f) Octet 7 and following shall specify the liveness-status of up to 240 DLEs, one bit per DLE, in order of descending NODE DL-address, encoded as an array of Booleans: 0 (DLE is not communicating and presumed "dead"), 1 (DLE is communicating and thus "alive").

The DLEs not expected to be connected to the link, as specified by the variables V(FUN) and V(NUN), are not included in this status. The array of Booleans shall be ordered so that bit 7 of octet 7 specifies the status of any DLE associated with the highest address (FF_{16}), bit 6 shall specify the status of any DLE associated with the next highest address (FE_{16}), and so on. The unused bits of the last octet shall be set to 0.

g) This field specifies the FDC status for up to 240 DLEs, one bit per DLE, in order of descending NODE address, encoded as an array of Booleans: 0 (DLE is not an FDC DLE) or 1 (DLE is an FDC DLE). The DLEs not expected to be connected to the link, as specified by the variables V(FUN) and V(NUN), are not included in this status. The array of Booleans shall be ordered so that bit 7 of first octet of this field specifies the status of any DLE associated with the highest address (FF_{16}), bit 6 shall specify the status of any DLE associated with the next highest address (FE_{16}), and so on. The DLE-type for any DLE with a liveness-status of 0, as indicated by the corresponding Boolean encoded in (f), shall be encoded as 0.

Trailing octets with the value zero shall be omitted from this field and from the SPDU. Therefore, when the local link has no FDC DLEs, then this entire field shall be omitted from the SPDU.

B.3.2.7 DL-conformance-query SPDU

The DL-conformance-query SPDU requests the receiving DLE to send a DL-conformance SPDU specifying the DL-conformance classes of the replying DLE to the requesting DLE. Its purposes are

a) to permit a new LAS DLE's DL-management to assess the impact of DLEs which are already-connected to the local link, facilitating recovery of the LAS role after failure of the previous LAS DLE, and to facilitate potential transfer of the LAS role to a more appropriate DLE;

b) to permit remote DL-management to interrogate DLEs on the extended link, by providing a query (and reply) which can be forwarded through a bridge; and

c) to facilitate DL-protocol analyzer initialization after attachment to an operating local link.

The DL-conformance-query SPDU shall be encoded with the format specified in Table B.1, as follows:

a) The SPDU header octet, octet 1, shall be encoded as 1110 1100, specifying a DL-conformance-query SPDU.

B.3.2.8 DL-conformance-reply SPDU

The DL-conformance-reply SPDU shall be formatted as specified in Table B.12. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.12 — DL-conformance-reply SPDU

octet index	contents of field
1	0010 0VVV
2	DL-protocol versions supported
3 **…** **6**	DL-conformance encoding

a) The SPDU header octet, octet 1, shall be encoded as 0010 0VVV, which identifies this as a DL-conformance-reply SPDU.

Table B.13 — DL-protocol versions supported

DL-protocol versions supported							
V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0
7	6	5	4	3	2	1	0

b) Octet 2 shall specify the version(s) of the DL-protocol supported, as depicted in Table B.13, where each V_K shall be encoded as a Boolean indicating whether protocol version K is supported.

Table B.14 — DL-conformance encoding (part 1)

supported connection-less extended data delivery features		**supported DL-priorities**	**timeliness time-stamp support**	**supported types of timeliness**			
remote-DLE-confirmed unitdata transfer	unitdata exchange	multi-priority	DL-time-stamped DLSDU support	TRANSPARENT	RESIDENCE	UPDATE	SYNCHRONIZED
C	X	M	D	T	R	U	S
7	6	5	4	3	2	1	0

c) Octet 3 shall specify, as depicted in Table B.14:

1) the DLE's support for connectionless-mode extended data-delivery features, encoded as a set of two Booleans;

2) the DLE's support for multiple DL-priorities, encoded as a Boolean;

3) the DLE's support for buffer time-stamps on DLCs, encoded as a Boolean; and

4) the DLE's supported types of DLC timeliness computation, encoded as a set of four Booleans.

Table B.15 — DL-conformance encoding (part 2)

subscribers-to-publisher support	**supported DLCEP data-delivery features**		**supported maximum DLSDU : DLPDU data ratio**				
X	FF		RRRRR				
7	6	5	4	3	2	1	0

d) Octet 4 shall specify, as depicted in Table B.15:

1) the DLE's support for the subscribers-to-publisher direction of multi-peer DLSDU transfer, X, as specified in 11.2(c) and Annex E/E.8.2.3, which shall be encoded as a Boolean;

2) the DLE's supported DLCEP data-delivery features, FF, as specified in 11.2(b) and Annex E/E.8.2.2, which shall be encoded as one less than the relevant item number of Annex E/E.8.2.2, with a value of 0 through 3; and

3) the DLE's maximum supported DLSDU : DLPDU data ratio for ORDERED DLCEPs, RRRRR, as specified in 11.2(d) and Annex E/E.8.2.4, which shall be encoded as the relevant supported value from Annex E/E.8.2.4, with a value of 0 through 16.

Table B.16 — DL-conformance encoding (part 3)

reserved				**supported time-based scheduling services**	**supported DL-time synchronism class**		
000				S	TTT		
7	6	5	4	3	2	1	0

e) Octet 5 shall specify, as depicted in Table B.16:

1) a four-bit field, coded as zero, reserved for future standards use;

2) the DLE's support for time-based scheduling services, S, as specified in 11.3(b) and Annex E/E.8.3.2, which shall be encoded as a Boolean; and

3) the DLE's DL-time synchronism class, TTT, as specified in 11.3(a) and Annex E/E.8.3.1, which shall be encoded as specified in 8.6(b.1).

All other values for these fields are invalid and shall not be used.

Table B.17 — DL-conformance encoding (part 4)

supported DLE class		**support for LONG DL-addresses**	**fractional duty cycle operation**	**token circulation desired**	**supported LAS time-based scheduling activity class**		
CC		A	F	D	SSS		
7	6	5	4	3	2	1	0

f) Octet 6 shall specify, as depicted in Table B.17:

1) the DLE's current class of operation, CC, as specified in 5.6 and Annex E/E.9.1, which shall be encoded as the relevant item number of Annex E/E.9.1, with a value of 1 through 3;

2) the DLE's support for LONG DL-addresses, encoded as a Boolean;

3) whether the DLE is an FDC DLE, F, which can be expected to be non-responsive to some live-list link-maintenance queries, encoded as 0 (not an FDC DLE) or 1 (an FDC DLE);

4) the DLE's need for token circulation without an explicit request, D, encoded as 0 (token circulation desired) or 1 (token circulation not desired); and

5) the DLE's support for LAS time-based scheduling activities, SSS, as specified in 11.3(c) and Annex E/E.8.3.3, which shall be encoded as the relevant item number of Annex E/E.8.3.3, with a value of 1 through 5;

All other values for these fields are invalid and shall not be used.

B.3.2.9 Link-basic-parameters-request SPDU

The link-basic-parameters-request SPDU requests the link-basic-parameters SPDU from the addressed LAS. The link-basic-parameters-request SPDU shall be encoded as specified in Table B.1:

a) The SPDU header octet, octet 1, shall be encoded as 1110 1101, specifying a link-basic-parameters-request SPDU.

B.3.2.10 Link-basic-parameters-reply SPDU

The link-basic-parameters-reply SPDU specifies the DL-configuration parameters for basic-class DLEs on the receiving DLE's link. The link-basic-parameters-reply SPDU shall be formatted as specified in Table B.18. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.18 — Link-basic-parameters-reply SPDU

octet index	contents of field
1	0010 1VVV
2 **…** **10**	same as octets 2…10 of node-activation SPDU
11 **12**	value for V(ST)
13	value for V(MRD)
14	value for V(MID)

a) The SPDU header octet, octet 1, shall be encoded as 0010 1VVV, which identifies this as a link-basic-parameters-reply SPDU.

b) Octets 2 to 10 shall be identical to the octets 2 to 10 of the node-activation SPDU (see Annex B/B.3.2.2).

c) Octets 11 and 12 shall specify the value for the receiving DLE's slot-time variable, V(ST), as specified in 5.7.1.1.

d) Octet 13 shall specify the value for the receiving DLE's maximum-reply-delay variable, V(MRD), as specified in 5.7.1.3.

e) Octet 14 shall specify the value for the receiving DLE's minimum-inter-DLPDU-delay variable, V(MID), as specified in 5.7.1.12. When the appropriate value for this variable is unknown, a value of zero shall be used.

B.3.2.11 Link-master-parameters-request SPDU

The link-master-parameters-request SPDU requests the link-parameters SPDU from the addressed LAS. The link-master-parameters-request SPDU shall be encoded as specified in Table B.1:

a) The SPDU header octet, octet 1, shall be encoded as 1110 1110, specifying a link-master-parameters-request SPDU.

B.3.2.12 Link-master-parameters-reply SPDU

The link-master-parameters-reply SPDU specifies the DL-configuration parameters for link-master-class DLEs on the receiving DLE's link. Its primary purpose is to permit link-master DLEs to prepare to assume the role of LAS on the local link. The link-master-parameters-reply SPDU shall be formatted as specified in Table B.19. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.19 — Link-master-parameters-reply SPDU

octet index	contents of field
1	0011 0VVV
2	value for V(MST)
3 **4**	value for V(DMDT)
5 **6**	value for V(DTHT)
7 **8**	value for V(LTHT)
9 **10**	value for V(TTRT)
11	value for V(FUN)
12	value for V(NUN)
13 **14**	value for V(TDP)
15 **16**	value for V(MICD)
17 **18**	value for V(LDDP)
19 **20**	value for V(ML)

a) The SPDU header octet, octet 1, shall be encoded as 0011 0VVV, which identifies this as a link-master-parameters-reply SPDU.

b) Octet 2 shall specify the value for the receiving DLE's local link's maximum-scheduled-traffic vaiable, V(MST), as specified in 5.7.5.5.

c) Octets 3 and 4 shall specify the value for the receiving DLE's local link's default-minimum-token-delegation-time variable, V(DMDT), as specified in 5.7.5.7.

d) Octets 5 and 6 shall specify the value for the receiving DLE's local link's default-token-holding-time variable, V(DTHT), as specified in 5.7.5.8.

e) Octets 7 and 8 shall specify the value for the receiving DLE's local link's link-maintenance-token-holding-time variable, V(LTHT), as specified in 5.7.5.9.

f) Octets 9 and 10 shall specify the value for the receiving DLE's local link's target-token-rotation-time variable, V(TTRT), as specified in 5.7.5.11.

g) Octet 11 shall specify the value for the receiving DLE's local link's first-unpolled-node id variable, V(FUN), as specified in 5.7.5.15.

h) Octet 12 shall specify the value for the receiving DLE's local link's number-of-consecutive-unpolled-nodes ids variable, V(NUN), as specified in 5.7.5.16.

j) Octets 13 and 14 shall specify the value for the receiving DLE's local link's time-distribution-period variable, V(TDP), as specified in 5.7.1.18. When the appropriate value for this variable is unknown, the minimum required value for the LAS's time-synchronism class (see 11.3(a)) shall be used.

k) Octets 15 and 16 shall specify the value for the receiving DLE's local link's maximum-inactivity-to-claim-LAS-delay variable, V(MICD), as specified in 5.7.5.19.

l) Octets 17 and 18 shall specify the value for the receiving DLE's local link's LAS-data-base-distribution-period variable, V(LDDP), as specified in 5.7.1.20.

m) Octets 19 and 20 shall specify the value for a receiving bridge DLE's maximum-link variable, V(ML), as specified in 5.7.6.1. Non-bridge DLEs which are functioning as an LAS DLE may specify a value of zero for this field of the SPDU when they do not maintain the proper value of V(ML) for the local link.

B.3.2.13 Token-hold-time-request SPDU

The token-hold-time-request SPDU requests the token-hold-time-array SPDU from the addressed LAS. The token-hold-time-request SPDU shall be encoded as specified in Table B.20:

Table B.20 — Token-hold-time-request SPDU

octet index	contents of field
1	1110 1111
2	starting NODE DL-address

a) The SPDU header octet, octet 1, shall be encoded as 1110 1111, specifying a token-hold-time-request SPDU.

b) Octet 2 shall specify the NODE DL-addresses for which maximum-token-hold-time is requested. The lower quartet of this octet shall always be 0. If maximum-token-hold-time for all DLEs is requested, then the value of this field shall be 00. If the value of this field is $N0_{16}$, where

N = 1 to F_{16}, then the request is for DLEs with NODE DL-addresses from $N0_{16}$ to NF_{16}.

B.3.2.14 Token-hold-time-array SPDU

The token-hold-time-array SPDU specifies the maximum-token-hold-time-array parameters (see 5.7.5.10) for DLEs on the receiving DLE's link. Its primary purpose is to permit link-master DLEs to prepare to assume the role of LAS on the local link. This SPDU is sent by the LAS either to all link master DLEs to update their database, or to a specific link master DLE as a response to token-hold-time-request from that link master DLE. The token-hold-time-array SPDU shall be formatted as specified in Table B.21. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.21 — Token-hold-time-array SPDU

octet index	contents of field
1	0011 1VVV
2	starting NODE DL-address
3 **4**	maximum-token-hold-time
33 **34**	maximum-token-hold-time

a) The SPDU header octet, octet 1, shall be encoded as 0011 1VVV, which identifies this as a token-hold-time-array SPDU.

b) Octet 2 shall specify the NODE DL-addresses for which maximum-token-hold-time is included in this SPDU. The lower quartet of this octet shall always be 0. If the value of this field is $N0_{16}$, where N = 1 to F_{16}, then this SPDU includes the maximum-token-hold-time for DLEs with NODE DL-addresses from $N0_{16}$ to NF_{16}. If the value of this field is 00, then this SPDU does not include the maximum-token-hold-time for any DLE, and it is used to indicate that the LAS has sent all of token-hold-time-array SPDUs, so that the receiving DLE can do error recovery, if it had not received all of such SPDUs.

c) Octets 3 to 34 shall specify the value of maximum-token-hold-time parameter for DLEs as an array of two octet values in the ascending sequence of NODE DL-addresses starting with $N0_{16}$ and ending with NF_{16}. If any DLE in this array is constructed or configured so that it does not need to receive a token delegated by a PT DLPDU, so that the value of the N subfield in B.3.2.1(c.1) was 1, then the maximum-token-hold-time for that DLE shall be set to zero.

B.3.2.15 FDC-DLE-has-“awakened” SPDU

The FDC-DLE-has-“awakened” SPDU notifies the local LAS DLE that the specified DLE is now online and ready to receive any DLPDUs which the LAS DLE has stored, awaiting the FDC DLE’s return to active participation on the local link.

The FDC-DLE-has-“awakened” SPDU shall be encoded with the format specified in Table B.1, as follows:

a) The SPDU header octet, octet 1, shall be encoded as 1111 1000, specifying an FDC-DLE-has-“awakened” SPDU.

B.3.2.16 FDC-DLE-may-“go-to-sleep”-notification SPDU

The FDC-DLE-may-“go-to-sleep”-notification SPDU notifies the local LAS DLE that the specified DLE is now ready to go temporarily offline, and that the LAS DLE should act as a bridge with respect to any non-WK DLPDUs addressed to the FDC DLE until the FDC DLE next sends an FDC-DLE-has-“awakened” SPDU. Upon receipt of this SPDU, the LAS DLE shall send an FDC-DLE-may-“go-to-sleep”-acknowledge SPDU to the requesting DLE.

The FDC-DLE-may-“go-to-sleep”-notification SPDU shall be encoded with the format specified in Table B.1, as follows:

a) The SPDU header octet, octet 1, shall be encoded as 1111 1001, specifying an FDC-DLE-may-“go-to-sleep”-notification SPDU.

B.3.2.17 FDC-DLE-may-“go-to-sleep”-acknowledge SPDU

The FDC-DLE-may-“go-to-sleep”-acknowledge SPDU notifies the requesting DLE that the local LAS DLE is prepared to act as a bridge with respect to any non-WK DLPDUs addressed to the FDC DLE until the FDC DLE next sends an FDC-DLE-has-“awakened” SPDU. Upon receipt of this SPDU, the FDC DLE may become inattentive to all non-WK DLPDUs received on the local link.

The FDC-DLE-may-“go-to-sleep”-acknowledge SPDU shall be encoded with the format specified in Table 1, as follows:

a) The SPDU header octet, octet 1, shall be encoded as 1111 1010, specifying an FDC-DLE-may-“go-to-sleep”-acknowledge SPDU.

B.3.3 LAS-transfer SPDUs

LAS-transfer SPDUs are used to coordinate link schedule information on a local link, and to transfer the LAS role from one DLE to another DLE on the same link.

B.3.3.1 Relinquish-LAS-role-request SPDU

This SPDU requests that the current LAS DLE transfer its role as LAS to the requesting DLE. If the current LAS DLE accepts the request, it shall transfer the LAS role by sending a TL DLPDU addressed to that requesting DLE as specified in 7.20.3. The relinquish-LAS-role-request SPDU shall be encoded with the format specified in Table B.1, as follows:

a) The SPDU header octet, octet 1, shall be encoded as 1111 0100, specifying a relinquish-LAS-role-request SPDU.

B.3.3.2 Accept-LAS-role-request SPDU

This SPDU requests that the receiving DLE prepare to accept the role as LAS from the requesting LAS DLE. If the addressed DLE accepts the request, it shall respond with an accept-LAS-role-reply SPDU, and shall take whatever actions are necessary to prepare for the transfer of the LAS role. When the addressed DLE is ready to accept the LAS role, it shall initiate the transfer by sending a relinquish-LAS-role-request SPDU (see B.3.3.1) to the LAS DLE. The accept-LAS-role-request SPDU shall be encoded with the format specified in Table B.1, as follows:

a) The SPDU header octet, octet 1, shall be encoded as 1111 0101, specifying an accept-LAS-role-request SPDU.

B.3.3.3 Accept-LAS-role-reply SPDU

This SPDU is sent in response to an accept-LAS-role-request SPDU (see B.3.3.2). By sending the response, the sending DLE accepts the request to become the LAS in the near future and agrees to take whatever actions are necessary (such as LAS-database synchronization) to prepare for the transfer of the LAS role. When the sending DLE is ready to accept the LAS role, it shall initiate the transfer by sending a relinquish-LAS-role-request SPDU (see B.3.3.1) to the LAS DLE. The accept-LAS-role-reply SPDU shall be encoded with the format specified in Table B.1, as follows:

a) The SPDU header octet, octet 1, shall be encoded as 1111 0110, specifying an accept-LAS-role-reply SPDU.

B.3.4 Schedule-construction SPDUs

Schedule-construction SPDUs convey requests from a non-LAS DLE to the LAS DLE on the same link, and replies from that LAS DLE to the requesting DLE. They can also be used to cancel previously-requested scheduling actions.

B.3.4.1 Sequence description encoding

Sequence descriptions are used in schedule construction (see B.3.4.2) SPDUs to define desired elements of LAS activity. Each sequence description consists of a series of encoded sequence elements, with the order of sequence elements the same as that in the defining DLS-user-specified sequence.

Each encoded sequence element consists of a header, as specified in Table B.22, followed by two or more octets of associated information.

Table B.22 — Sequence element header encoding

<table>
<tr><th colspan="6">item type and format</th><th colspan="2">priority</th></tr>
<tr><td colspan="6">TTTTTT</td><td colspan="2">PP</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

The sequence-element header shall encode the type and format, and priority if relevant, of the sequence element. When present, the priority field shall be encoded as specified in 6.2.1.3.

B.3.4.1.1 SHORT DL-address and duration sequence element

Table B.23 — SHORT DL-address and duration sequence element

octet index	contents of field
1	TTTT TTPP
2 **3**	SHORT DL-address
4 **5**	duration

A SHORT DL-address and duration sequence element shall be encoded as specified in Table B.23:

a) The sequence element header octet, octet 1, shall be encoded as TTTT TTPP, in accordance with Table B.22, as follows:

01_{16} — CD-request, SHORT DL-address and associated duration, URGENT priority;

02_{16} — CD-request, SHORT DL-address and associated duration, NORMAL priority;

03_{16} — CD-request, SHORT DL-address and associated duration, TIME-AVAILABLE priority; and

04_{16} — ES-request, SHORT DL-address and associated duration;

b) Octets 2 and 3 of the sequence element shall specify a SHORT DLCEP-address (CD-request) or SHORT DLSEP-address (ES-request).

c) Octets 4 and 5 shall specify a two-octet maximum duration of reply-token (CD-request) or delegated-token (ES-request) usage, measured in octets of link transmission capacity.

This is the maximum duration of a single instance of usage of the specified token by the addressed DLE, and does not include the link capacity required to send CD or ES DLPDU, immediate-response-recovery-delay slot-times, V(IRRD) $\times$ V(ST) octet-durations and the retry considerations of the LAS DLE in sending the token through transmission of the CD or ES DLPDU.

B.3.4.1.2 LONG DL-address and duration sequence element

Table B.24 — LONG DL-address and duration sequence element

octet index	contents of field
1	TTTT TTPP
2 **3** **4** **5**	LONG DL-address
6 **7**	duration

A LONG DL-address and duration sequence element shall be encoded as specified in Table B.24:

a) The sequence element header octet, octet 1, shall be encoded as TTTT TTPP, in accordance with Table B.22, as follows:

09_{16} — CD-request, LONG DL-address and associated duration, URGENT priority;

$0A_{16}$ — CD-request, LONG DL-address and associated duration, NORMAL priority;

$0B_{16}$ — CD-request, LONG DL-address and associated duration, TIME-AVAILABLE priority; and

$0C_{16}$ — ES-request, LONG DL-address and associated duration;

b) Octets 2 through 5 of the sequence element shall specify a delocalized LONG DLCEP-address (CD-request) or LONG DLSEP-address (ES-request).

c) Octets 6 and 7 shall specify a two-octet maximum duration of reply-token (CD-request) or delegated-token (ES-request) usage, measured in octets of link transmission capacity.

This is the maximum duration of a single instance of usage of the specified token by the addressed DLE, and does not include the link capacity required to send CD or ES DLPDU, immediate-response-recovery-delay slot-times, V(IRRD) × V(ST) octet-durations and the retry considerations of the LAS DLE in sending the token through transmission of the CD or ES DLPDU.

B.3.4.1.3 Wakeup request sequence element

Table B.25 — Wakeup request sequence element

octet index	contents of field
1	TTTT TT00
2	NODE DL-address

A wakeup request sequence element shall be encoded as specified in Table B.25:

a) The sequence element header octet, octet 1, shall be encoded as TTTT TT00, in accordance with Table B.22, as follows:

00_{16} — WK-request, NODE DL-address;

b) Octet 2 of the sequence element shall specify a NODE DL-address.

NOTE — While a Wakeup sequence element cannot be included in a DLS-user-originated schedule request, it can be included in a scheduling request made by a fractional-duty-cycle (FDC) DLE wishing to schedule its own “reawakening”.

B.3.4.2 Schedule-request SPDU

Scheduling requests consist of requests for either one-time or periodic (that is, cyclic with a fixed period) execution of a sequence, subject to specified scheduling constraints. These requests differ only in the parameters specifying the required period and permissible jitter of cyclic execution of the sequence, which are not relevant to one-time execution of the sequence.

Table B.26 — Schedule-request SPDU

octet index	contents of field
1	0110 0VVV
2	Sequence type, schedule type, and schedule priority
3 **4**	Schedule identifier
5 **6**	value for V_S(TSL)
7 **8** **9** **10** **11** **12**	Desired starting time
13 **14** **15** **16**	Earliest starting time, relative to desired starting time
17 **18** **19** **20**	Latest ending time, relative to desired starting time
21 **22** **23** **24**	Period of cyclic sequence
25 **26** **27** **28**	Maximum permissible jitter
29 **…**	Sequence description

Scheduling-request SPDUs shall be formatted as specified in table B.26:

a) The SPDU header octet, octet 1, shall be encoded as 0110 0VVV.

Table B.27 — Sequence type, schedule type and priority encoding

sequence type			**schedule type**			priority	
SSS			TTT			PP	
7	6	5	4	3	2	1	0

b) Octet 2 shall identify the sequence type, the schedule type, and the priority of the scheduling request. It shall be encoded as specified in Table B.27:

1) The sequence type SSS shall specify whether the entire sequence is to be executed contiguously or whether discontiguous (interrupted) execution is permitted, and in the latter case whether the sequence as a whole is to be considered ordered or unordered:

— A **contiguous** sequence is one in which the DLS-provider must schedule enough consecutive segments of local-link capacity to execute the entire sequence.

— A **discontiguous** sequence is one in which the DLS-provider may intersperse the execution of the sequence with other link activity.

— An **ordered** sequence is one in which the DLS-provider must execute the sequence elements in the order specified.

— An **unordered** sequence is one in which the DLS-provider may rearrange the order of execution of the elements of the sequence, facilitating DLL-internal optimizations.

NOTE — Contiguous sequences are always treated as ordered, since unordered permits no additional optimization.

This subfield shall be encoded in three bits as:

000) unordered discontiguous;

001) ordered discontiguous; and

011) (ordered) contiguous.

All other values are reserved for future standards use.

2) The schedule type TTT shall be encoded in three bits as:

001) one-time; and

100) periodic.

All other values are reserved for future standards use.

3) The priority PP shall be encoded in two bits as specified in 6.2.1.3.

c) Octets 3 and 4 shall encode a non-zero schedule-identifier provided by the requesting DLE.

NOTE — This identifier enables the DLE which requests scheduling to associate a reply with its corresponding scheduling request.

d) Octets 5 and 6 shall encode the current value of the DL-time source link-id variable, V_S(TSL), of the sending DLE.

e) Octets 7 through 12 shall encode the desired starting DL-time for the first instance of execution of the scheduled sequence, in units of 2^{-5} ms, where that time is referenced to the DL-time source link-id specified in (d).

f) Octets 13 through 16 shall encode the offset of the earliest acceptable starting DL-time for the first instance of execution of the scheduled sequence, in units of 2^{-5} ms, where that time is referenced to the DL-time specified in octets 7 through 12.

g) Octets 17 through 20 shall encode the offset of the latest acceptable ending DL-time for the first instance of execution of the scheduled sequence, in units of 2^{-5} ms, where that time is referenced to the DL-time specified in octets 7 through 12.

h) In periodic-scheduling SPDUs, octets 21 through 24 shall encode the repetition period for execution of a cyclic scheduled sequence, in units of 2^{-5} ms. In one-time-scheduling SPDUs, octets 21 through 24 shall encoded as zero.

j) In periodic-scheduling SPDUs, octets 25 through 28 shall encode the maximum permissible cycle-to-nominal jitter, in units of 2^{-5} ms, in the repetitive starting times of a cyclic scheduled sequence. In one-time-scheduling SPDUs, octets 25 through 28 shall encoded as zero.

k) The remaining octets, through the last octet of the SPDU, shall encode a description of the sequence to be scheduled, as specified in B.3.4.1.

The length of this SPDU may approach 256 octets.

B.3.4.3 Scheduling-completed SPDU

The LAS DLE shall acknowledge receipt and completion of a schedule-request SPDU by replying with a scheduling-completed SPDU, which shall be encoded as specified in Table B.28:

a) The SPDU header octet, octet 1, shall be encoded as 0110 1VVV, specifying a scheduling-completed SPDU.

Table B.28 — Scheduling-completed SPDU

octet index	contents of field
1	0110 1VVV
2 **3**	Schedule identifier
4	Status or reason
5 **6**	value for V_S(TSL)
7 **8** **9** **10** **11** **12**	Scheduled starting time

b) Octets 2 and 3 shall encode the schedule-identifier provided by the requesting DLE in the schedule-request SPDU to which this SPDU is a reply.

NOTE — This identifier enables the DLE which requested the scheduling to associate the reply with its corresponding scheduling request.

c) Octet 4 shall encode the status of the corresponding request — either success or the reason for failure, as specified in Table B.29.

Table B.29 — Status and reason codes

SUCCESS	00_{16}
SERVICE-UNAVAILABLE — TEMPORARY	02_{16}
RESOURCE-UNAVAILABLE — TEMPORARY	03_{16}
TIME-BASE MISMATCH — TEMPORARY	04_{16}
SERVICE-UNAVAILABLE — PERMANENT	82_{16}
RESOURCE-UNAVAILABLE — PERMANENT	83_{16}

d) If octet 4 encodes success (00), then octets 5 and 6 shall encode the current value of the DL-time source link-id variable, V_S(TSL), of the LAS DLE.

e) If octet 4 encodes success (00), then octets 7 through 12 shall encode the scheduled starting DL-time for execution of the sequence, in units of 2^{-5} ms, where that time is referenced to the DL-time source link-id specified in (d).

f) If octet 4 encodes failure (any value other than 00), then this SPDU shall have only 4 octets and the next two fields shall be omitted.

B.3.4.4 Cancel-schedule SPDU

Table B.30 — Cancel-schedule SPDU

octet index	contents of field
1	1111 1100
2 **3**	Schedule identifier

The DLE which requested a scheduling action of the LAS DLE may also request the cancellation of that previous scheduling action. The SPDU by which this request is made shall be encoded as specified in Table B.30:

a) The SPDU header octet, octet 1, shall be encoded as 1111 1100, specifying a cancel-schedule SPDU.

b) Octets 2 and 3 shall encode the schedule-identifier provided by the requesting DLE in the schedule-request SPDU which this SPDU is canceling.

NOTE — This identifier enables the LAS DLE to associate the cancellation request with the corresponding scheduling request.

B.3.4.5 Schedule-canceled SPDU

Table B.31 — Schedule-canceled SPDU

octet index	contents of field
1	1111 1101
2 **3**	Schedule identifier
4	Status or reason

The LAS DLE shall acknowledge cancellation of a scheduled sequence by sending a schedule-canceled SPDU to the DLE which originally made the schedule request. The schedule-canceled SPDU shall be encoded as specified in Table B.31:

a) The SPDU header octet, octet 1, shall be encoded as 1111 1101, specifying a schedule-canceled SPDU.

b) Octets 2 and 3 shall encode the schedule-identifier provided by the requesting DLE in the schedule-request SPDU which this SPDU is canceling.

NOTE — This identifier enables the DLE which requested the scheduling to associate the reply with its corresponding scheduling request.

c) Octet 4 shall encode the reason for cancellation, as specified in Table B.29.

B.3.5 Schedule-transfer SPDUs

Schedule-transfer SPDUs convey link schedule from LAS DLE to a non-LAS DLE on the same link. This transfer allows another link master DLE to assume LAS role and continue the execution of the link schedule. The link schedule transferred via these SPDUs could have been constructed by the LAS or could have been originally transferred to the LAS from another LAS DLE or from the DL-management. The format of these SPDUs does not save the original schedule-request SPDU. Therefore, a LAS DLE, which received such schedule from another LAS DLE or from the DL-management cannot modify this schedule.

The Link schedule consists of one or more periodic sub-schedule SPDUs and one schedule-summary SPDU as shown in Table B.32. Each sub-schedule contains one or more sequences, all of the same execution period. A LAS may have a limit on the number of different sub-schedules it can execute simultaneously. The schedule can be transferred one SPDU at a time, or all SPDUs can be included in one DLPDU. If the schedule is transferred in one DLPDU, then the SPDUs should be in one contiguous sequence as shown in the Table B.32. The format of each SPDU is such that the end of each SPDU can be determined without any ambiguity.

The 3-bit version number, NNN, of the schedule transfer encoding which is used in schedule transfer SPDUs shall be encoded as 000.

Table B.32 – Link-schedule

Schedule-summary SPDU
Sub-schedule SPDU
Sub-schedule SPDU

B.3.5.1 Schedule-summary SPDU

The schedule-summary SPDU is a record of attributes which are common to all sub-schedules. This SPDU shall be encoded as specified in Table B.33. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field. If this SPDU is for a null schedule, then it shall consist of only the first four octets, and the remaining octets shown in B.33 shall be omitted from the SPDU:

a) The SPDU header octet, octet 1, shall be encoded as 1111 1111, specifying schedule transfer SPDU.

b) Octet 2 shall specify the type of schedule transfer SPDU — schedule-summary, and shall be encoded as 0000 0NNN, where NNN is the version number of the schedule transfer encoding.

Table B.33 – Schedule-summary SPDU

octet index	contents of field
1	1111 1111
2	0000 0NNN
3 **4**	schedule version-number
5 **6**	schedule-builder identifier (0100…0FFF)
7	number of sub-schedules
8	V(MSO)
9 **10**	storage required for total schedule
11 **12**	timing resolution required to meet schedule-jitter commitments
13 **14**	V(MRD) ¥V(ST)
15 **16**	V(TSL)
17 **18** **19** **20** **21** **22**	periodic schedule DL-time base (T0)
23 **24** **25** **26**	macro-cycle duration
27 **28**	sub-schedule-SPDU reference
…	sub-schedule-SPDU reference
…	sub-schedule-SPDU reference
…	end-of-schedule-summary

c) Octets 3 and 4 shall specify the non-zero value of the schedule version-number as a two octet integer, or shall specify the value zero. If the value of this field is zero, which is used to indicate a null schedule, then this field shall be the last field actually present in the SPDU.

d) Octets 5 and 6 shall specify the source of the schedule as an identifier with a value in the range of 0100_{16} … $0FFF_{16}$.

NOTE — This field is only for reference and is not used by the LAS.

e) Octet 7 shall specify the number of sub-schedules in the schedule.

f) Octet 8 shall specify the value of V(MSO), maximum-scheduling-overhead (see 5.7.5.6), which was used in the construction of the schedule. Its unit of measurement and range of values is specified in 5.7.5.6.

g) Octets 9 and 10 shall specify the total storage space required for the schedule-summary and all of sub-schedules in number of octets. This is the sum of number of octets in each SPDU of the schedule.

h) Octets 11 and 12 shall specify the minimum resolution of the time clock required to execute this schedule and still meet the jitter specifications to which the schedule was constructed. Its units are 2^{-5} ms.

i) Octets 13 and 14 shall specify the value of maximum-response-delay slot-times, V(MRD) × V(ST) octet-durations, which was used in the construction of the schedule. The unit of measurement and range of values are specified in 5.7.1.1 and 5.7.1.3.

j) Octets 15 and 16 shall specify the link-id of the source of time distribution, which was used to specify the DL-time used in this schedule.

k) Octets 17 to 22 shall specify the starting DL-time (T0), referenced to the DL-time source link-id specified in (j), of all periodic sub-schedules in this schedule. The start time of all sequences in all sub-schedules is shown as offset with respect to this periodic schedule DL-time base. Its units are 2^{-5} ms.

l) Octets 23 to 26 shall specify the duration of macrocycle, which is equal to the least common multiple of periods of all sub-schedules. Its units are 2^{-5} ms.

m) Octets 27 and up to the end of the SPDU shall reference the sub-schedule-SPDUs by identifiers, as shown in Table B.34. Each identifier shall be 2 octets long and shall have a non-zero value.

n) The end-of-schedule-summary shall be encoded as a multiple of two octets whose values are all zero.

NOTE — The schedule builder can fill the end of this SPDU with more than the minimum number of octets in the anticipation of future addition of sub-schedule-SPDU references.

Table B.34 – Sub-schedule-SPDU reference

octet index	**contents of subfield**
1 **2**	sub-schedule identifier

The length of this SPDU may approach 256 octets.

B.3.5.2 Sub-schedule SPDU

The sub-schedule SPDU is a record of attributes which are common to one sub-schedule and specification of all sequences for that sub-schedule. This SPDU shall be encoded as specified in Table B.35. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.35 – Sub-schedule SPDU

octet index	contents of subfield
1	1111 1111
2	0000 1NNN
3 **4**	schedule version-number
5 **6**	sub-schedule identifier
7 **8** **9** **10**	sub-schedule period
11 **…**	sequence sub-SPDU
…	sequence sub-SPDU
…	sequence sub-SPDU
…	end-of-sub-schedule

a) The SPDU header octet, octet 1, shall be encoded as 1111 1111, specifying schedule transfer SPDU.

b) Octet 2 shall specify the type of schedule transfer SPDU - sub-schedule, and shall be encoded as 0000 1NNN, where NNN is the version number of the schedule transfer encoding.

c) Octets 3 and 4 shall specify the two-octet integer value of the version-number of the schedule which includes this sub-schedule; all values except zero are permitted. The value of this field shall be identical to the same field of schedule-summary SPDU.

d) Octets 5 and 6 shall either specify the non-zero value of a sub-schedule identifier as referenced in the schedule-summary SPDU (see B.3.5.1(p)) or it shall be zero. If the value of this field is zero, then this shall be the last octet of this SPDU, and it is used to indicate that the LAS has sent all of sub-schedule SPDUs, so that the receiving DLE can do error recovery, if it had not received all of such SPDUs. If the value of this octet is non-zero, then this octet shall be followed by the other fields of this SPDU as specified in (e), (f) and (g).

e) Octets 7 to 10 shall specify a non-zero execution period of all sequences in the sub-schedule. Its units are 2^{-5} ms.

f) Octets 11 and up to the end of the SPDU shall specify one or more sequences as specified in B.3.5.2.1.

g) The end-of-sub-schedule shall be encoded as a multiple of four octets whose values are all zero.

NOTE — The schedule builder can fill the end of this SPDU with more than the minimum number of octets in the anticipation of future addition of sequence sub-SPDUs.

The length of this SPDU may approach 256 octets.

B.3.5.2.1 Sequence sub-SPDU

The sequence sub-SPDU is a sub-record of attributes which are common to one sequence and specification of all elements of that sequence. This sub-SPDU shall be encoded as specified in Table B.36. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.36 – Sequence Sub-SPDU

octet index	contents of subfield
1 **2** **3** **4**	scheduled-starting-time-offset
5 **6**	Maximum computed duration of sequence execution
7 **…**	element-description
…	element-description
…	element-description
…	end-of-sequence

a) Octets 1 to 4 shall specify a non-zero starting time of the sequence as offset from periodic schedule DL-time base (T0). Its units are 2^{-5} ms. The LAS is required to execute the entire sequence continuously starting from the first to the last element. The starting time of the sequence is

$$T0 + N \times (\text{sub-schedule period}) + \text{scheduled-starting-time-offset},$$

where N is a non-negative integer.

NOTE — Whenever there is a change in the DL-time-offset, V(DLTO), the periodic schedule time-base, (T0), is adjusted by the same amount. Therefore, once the execution of a schedule has begun, the start time of each sequence is given by

$$V(LSTO) + C(NT) =$$

$$V_{start}(LSTO) + C_{start}(NT) + N \times (\text{sub-schedule-period}) + \text{scheduled-starting-time-offset},$$

where the sum $V_{start}(LSTO) + C_{start}(NT)$ is the value of V(LSTO) + C(NT) at the start of the schedule execution.

b) Octets 5 and 6 shall specify the worst case duration of executing the sequence. This field is only for reference and LAS does not use this to execute the sequence.

c) Octet 7 and following shall specify one or more elements as specified in B.3.5.2.2.

d) The end-of-sequence shall be encoded as two octets whose values are both zero.

B.3.5.2.2 Element

The element is the lowest level component of the schedule, and it represents a transaction (see 3.3.18). The element shall be encoded as specified in Table B.37. Multi-octet values shall be encoded with the most significant octet of the value encoded in the lowest-index octet of the multi-octet field:

Table B.37 – Element-description

octet index	contents of subfield
1	element-type
2 **…**	element-parameter

a) Octets 1 shall specify the type of transaction and shall be encoded as:

1011 00PP — CD-request, SHORT DL-address, PP = priority (see 6.2.1.3);

1011 10PP — CD-request, LONG DL-address, PP = priority (see 6.2.1.3);

1000 0000 — ES-request, SHORT DL-address, first and last ES of one sequence;

1000 0010 — ES-request, SHORT DL-address, first ES of one sequence;

1000 0110 — ES-request, SHORT DL-address, subsequent, but not last ES of one sequence;

1000 0111 — ES-request, SHORT DL-address, last ES of one sequence;

1000 1000 — ES-request, LONG DL-address, first and last ES of one sequence;

1000 1010 — ES-request, LONG DL-address, first ES of one sequence;

1000 1110 — ES-request, LONG DL-address, subsequent, but not last ES of one sequence;

1000 1111 — ES-request, LONG DL-address, last ES of one sequence; and

0000 0000 — WK-request, NODE DL-address;

b) Octet 2 and up shall specify the parameters for the element. The length and encoding depends upon the element-type:

1) If element type is CD-request, SHORT DL-address, then the element parameter shall have two octets and these two octets, octet 2 and 3 shall specify a SHORT DLCEP-address.

2) If element type is CD-request, LONG DL-address, then the element parameter shall have four octets and these four octets, octet 2 to 5 shall specify a LONG DLCEP-address.

3) If the element type is ES-request, SHORT DL-address, then the element parameter shall have four octets. The first two octets of these four octets, octet 2 and 3 shall specify a SHORT DLSEP-address. The next two octets of these four octets, octet 4 and 5 shall specify duration measured in octets of link transmission capacity, DD-parameter of ES DLPDU.

4) If the element type is ES-request, NODE DL-address, then the element parameter shall have six octets. The first four octets of these four octets, octet 2 to 5 shall specify a NODE DLSEP-address. The next two octets of these six octets, octet 6 and 7 shall specify duration measured in octets of link transmission capacity, DD-parameter of ES DLPDU.

5) If the element type is WK-request, NODE DL-address, then the element parameter shall have one octet and this octet, octet 2 shall specify a non-zero NODE DL-address.

B.3.5.3 Schedule-summary-request SPDU

The schedule-summary-request SPDU requests the schedule-summary SPDU for the currently active schedule, from the addressed LAS. The schedule-summary-request SPDU shall be encoded as specified in Table B.38:

Table B.38 — Schedule-summary-request SPDU

octet index	contents of field
1	1111 1110
2	0000 0000

a) The SPDU header octet, octet 1, shall be encoded as 1111 1110, specifying a schedule transfer request SPDU.

b) Octet 2 shall be encoded as 0000 0000, specifying a schedule-summary-request SPDU.

B.3.5.4 Sub-schedule-request SPDU

The sub-schedule-request SPDU requests the sub-schedule SPDU from the addressed LAS. The sub-schedule-request SPDU shall be encoded as specified in Table B.39:

Table B.39 — Sub-schedule-request SPDU

octet index	contents of field
1	1111 1110
2	0000 1000
3 **4**	schedule version-number
5 **6**	sub-schedule identifier

a) The SPDU header octet, octet 1, shall be encoded as 1111 1110, specifying a schedule transfer request SPDU.

b) Octet 2 shall be encoded as 0000 1000, specifying a sub-schedule-request SPDU.

c) Octets 3 and 4 shall specify the two octet integer value of the version-number of the schedule; all values except zero are permitted.

d) Octets 5 and 6 shall either specify the non-zero value of a sub-schedule identifier as referenced in the schedule-summary SPDU (see B.3.5.1(p)) or it shall be zero. If the value of this field is zero, then this is a request to LAS to send all sub-schedule SPDUs for the specified schedule, otherwise it is a request to LAS to send the specified sub-schedule SPDU.

B.3.6 Non-LAS SPDUs

Non-LAS SPDUs are used to support DL-services other than scheduling, such as DL-SUBSCRIBER-QUERY requests and DL-LISTENER-QUERY requests, to support bridge filtering database maintenance operations, and to support protocol-analyzers. These SPDUs may be sent between any two DLEs on the extended link, and may be broadcast to all DLEs on a local link where appropriate.

B.3.6.1 Parameter lists in SPDUs

Non-LAS SPDUs are used to query or transfer DLS-provider database information from one DLE to another. These SPDUs use a common parameter-list structure to encode the transferred database items, and to specify the beginning, continuation, or ending of such a list.

Each encoded element of parameter-list consists of a header, as specified in Table B.40, followed by zero or more octets of associated information.

Table B.40 — Parameter-list element-header encoding

list element type				list element subtype			
TTTT				SSSS			
7	6	5	4	3	2	1	0

The list element-header shall encode the type and subtype, if relevant, of the list element. The format and primary function of the list element shall be determined by the list element type, TTTT, whose encodings shall be:

0 begin/end-of-list element;

1 continuation-of-list element;

2 SHORT DL-address element;

3 LONG DL-address element;

4 DLSAP-address-characteristics element; and

5 DLCEP-characteristics element.

B.3.6.1.1 Begin/end-of-list element

Table B.41 — Begin/end-of-list element

octet index	contents of field
1	0000 0000

A begin/end-of-list element shall be encoded as specified in Table B.41:

a) The list element header octet, octet 1, shall be encoded as 0000 0000, in accordance with Table B.40.

B.3.6.1.2 Continuation-of-list element

Table B.42 — Continuation-of-list element

octet index	contents of field
1	0001 0000
2 **3** **4** **5**	Continuation identifier

A continuation-of-list element shall be encoded as specified in Table B.42:

a) The list element header octet, octet 1, shall be encoded as 0001 0000, in accordance with Table B.40.

b) Octets 2 through 5 shall encode a continuation-identifier provided by the DLE whose DLS-provider database is being read. There is no constraint on its value or interpretation.

NOTE — This identifier enables the DLE whose DLS-provider database is being read to pass a *handle* to its peer DLE, for later return in another database-accessing request, such that the two DLEs can continue the database access at later time using only the connectionless lower-level DL-protocol.

B.3.6.1.3 SHORT DL-address element

Table B.43 — SHORT DL-address list element

octet index	contents of field
1	0010 SSSS
2 **3**	SHORT DL-address

Each SHORT DL-address list element shall be encoded as specified in Table B.43:

a) The list element header octet, octet 1, shall be encoded as 0010 SSSS, in accordance with Table B.40, and the sub-type field, SSSS, shall specify the DL-address type of the accompanying DL-address parameter, as follows:

20_{16} — SHORT group DL-address;

21_{16} — SHORT DLSAP-address whose DL(SAP)-role is BASIC;

24_{16} — SHORT DLSAP-address whose DL(SAP)-role is INITIATOR;

25_{16} — SHORT DLSAP-address whose DL(SAP)-role is UNCONSTRAINED RESPONDER;

26_{16} — SHORT DLSAP-address whose DL(SAP)-role is CONSTRAINED RESPONDER;

28_{16} — SHORT DLCEP-address of a peer DLCEP;

29_{16} — SHORT DLCEP-address of a publisher DLCEP;

$2A_{16}$ — SHORT DLCEP-address of a subscriber DLCEP's publisher;

$2B_{16}$ — SHORT DLCEP-address of a subscriber DLCEP (which exists during DLCEP-establishment); and

$2C_{16}$ — SHORT DLSEP-address.

b) Octets 2 and 3 of the list element shall specify a SHORT DL-address.

B.3.6.1.4 LONG DL-address element

Table B.44 — LONG DL-address element

octet index	contents of field
1	0011 SSSS
2 **3** **4** **5**	LONG DL-address

Each LONG DL-address list element shall be encoded as specified in Table B.44:

a) The list element header octet, octet 1, shall be encoded as 0011 SSSS, in accordance with Table B.40, and the sub-type field, SSSS, shall specify the DL-address type of the accompanying DL-address parameter, as follows:

30_{16} — LONG group DL-address;

31_{16} — LONG DLSAP-address whose DL(SAP)-role is BASIC;

34_{16} — LONG DLSAP-address whose DL(SAP)-role is INITIATOR;

35_{16} — LONG DLSAP-address whose DL(SAP)-role is UNCONSTRAINED RESPONDER;

36_{16} — LONG DLSAP-address whose DL(SAP)-role is CONSTRAINED RESPONDER;

38_{16} — LONG DLCEP-address of a peer DLCEP;

39_{16} — LONG DLCEP-address of a publisher DLCEP;

$3A_{16}$ — LONG DLCEP-address of a subscriber DLCEP's publisher;

$3B_{16}$ — LONG DLCEP-address of a subscriber DLCEP (which exists during DLCEP-establishment); and

$3C_{16}$ — LONG DLSEP-address.

b) Octets 2 through 5 of the list element shall specify a delocalized LONG DL-address.

B.3.6.1.5 DLSAP-address-characteristics element

Table B.45 — DLSAP-address-characteristics element

octet index	contents of field
1	0100 0000
2 **…** **5**	constrained responder's remote-DLSAP-address as a LONG DL-address

Each DLSAP-address-characteristics list element shall be encoded as specified in Table B.45:

a) The list element header octet, octet 1, shall be encoded as 0100 0000, in accordance with Table B.40.

b) Octets 2 through 5 shall encode the remote DLSAP-address parameter of a DLSAP-address whose DL(SAP)-role binding is CONSTRAINED RESPONDER. This encoding shall be in the form of a delocalized LONG DL-address.

B.3.6.1.6 DLCEP-characteristics element

Table B.46 — DLCEP-characteristics element

octet index	contents of field
1	0101 0000
2 **3**	basic DLCEP parameters
4 **…** **7**	sender DLCEP parameters
8 **…** **11**	receiver DLCEP parameters

Each DLCEP-characteristics list element shall be encoded as specified in Table B.46:

a) The list element header octet, octet 1, shall be encoded as 0101 0000, in accordance with Table B.40.

b) Octets 2 and 3 shall encode the basic parameters of the DLCEP, as specified for the first and second octets of the parameters of an EC DLPDU in Tables 29 and 30 and in 8.1(a.2) through (a.8), except that reply-request subfield (R) (see 8.1(a.1)) shall be coded as 0.

c) Octets 4 through 7 shall encode the sender parameters of the DLCEP, as specified for the seventh through tenth octets, respectively, of an EC DLPDU in Tables 33 through 35 and in 8.1(c.1) through (c.10).

d) Octets 8 and 11 shall encode the receiver parameters of the DLCEP, as specified for the eleventh through fourteenth octets, respectively, of an EC DLPDU in Tables 36 through 38 and in 8.1(d.1) through (d.10).

B.3.6.2 Address-query SPDU

Table B.47 — Address-query SPDU

octet index	contents of field
1	0100 0VVV
2	query modifiers
3 **4** **5** **6**	request identifier
7 **…**	DL-address parameter-list element

The address-query SPDU requests the receiving DLE to search its forwarding database (of DL-addresses recognized for either local reception or forwarding) for the DL-address specified by the associated single DL-address parameter list element, and to compose and send an address-report SPDU if that address is found. Its primary purpose is to support DL-SUBSCRIBER-QUERY and DL-LISTENER-QUERY requests and link protocol analyzers.

The address-query SPDU shall be formatted as specified in Table B.47:

a) The SPDU header octet, octet 1, shall be encoded as 0100 0VVV, specifying an address-query SPDU.

b) Octet 2 specifies whether additional information on the DLSAP-address-binding or DLCEP should be contained within a responder's address-report SPDU. Its encoding shall be:

00_{16} — No accompanying peer DL-address or DLCEP information is requested.

01_{16} — Responders should include list elements for peer DL-address, and, where appropriate, for DLCEP-characteristics, in the replying address-report SPDU when known.

Other values are reserved.

c) Octets 3 through 6 shall encode a non-zero request-identifier provided by the requesting DLE.

NOTE — This identifier enables the requesting DLE to associate a reply with its corresponding request.

d) Octets 7 and following shall encode either one SHORT DL-address element or one LONG DL-address element, as specified in B.3.6.1.3 and B.3.6.1.4. The encoded DL-address shall specify the DL-address which is the subject of the query.

B.3.6.3 Address-report SPDU

Table B.48 — Address-report SPDU

octet index	contents of field
1	0100 1VVV
2	reason for report
3 **4** **5** **6**	request identifier
7 **…**	parameter-list elements

The address-report SPDU is sent in response to an address-query SPDU when a receiving DLE determines that the queried DL-address is contained within the receiving DLE's forwarding database, for either local reception or for forwarding to another DLE.

It is also sent to all local bridges to report the activation or deactivation of a DL-address

1) by a non-FDC DLE when the DL-address is non-local (that is, has a link component with a value other than zero or V(TL)); and

2) by an FDC DLE for all of its addresses.

NOTE — DL-address activation occurs upon executing a DL-BIND or DL-CONNECT request; DL-address deactivation occurs upon executing a DL-UNBIND or DL-DISCONNECT request.

Its primary purposes are to support DL-SUBSCRIBER-QUERY and DL-LISTENER-QUERY requests, and to facilitate bridges and link protocol analyzers.

The address-report SPDU may be addressed to a DLE on another link. It shall be formatted as specified in Table B.48:

a) The SPDU header octet, octet 1, shall be encoded as 0100 1VVV, specifying an address-report SPDU.

b) Octet 2 specifies the reason for the address-report SPDU; it shall be encoded as:

00_{16} — The SPDU is a reply to a prior address-query SPDU;

10_{16} — A non-FDC DLE has activated the specified DL-address;

11_{16} — An FDC DLE has activated the specified DL-address;

12_{16} — A non-FDC DLE has deactivated the specified DL-address;

13_{16} — An FDC DLE has deactivated the specified DL-address.

All other values are reserved for future standard use.

c) Octets 3 through 6 shall encode

1) the non-zero request-identifier provided by the requesting DLE in the address-query SPDU to which this SPDU is a reply; or

NOTE — This identifier enables the querying DLE to associate the reply with its corresponding query.

2) a value of zero indicating that this is a spontaneous report of the activation or deactivation of a DL-address, and not a reply to a prior SPDU.

d) Octets 7 and following shall encode either one SHORT DL-address element or one LONG DL-address element, as specified in B.3.6.1.3 and B.3.6.1.4.

e) If this SPDU is a reply to an address-query SPDU, and that associated query SPDU specified that responders should include peer DL-address and DLCEP-characteristics elements in the reply SPDU when known:

1) The responding DLE shall append a DLSAP-address-characteristics or DLCEP-address-characteristics element specifying the characteristics of the queried DL-address to the reply SPDU, as specified in B.3.6.1.5 and B.3.6.1.6; and then

2) if the queried DL-address specified a peer DLCEP-address, then the responding DLE shall append a DL-address element specifying the remote peer DL-address of the queried DL-address to the reply DLPDU, as specified in B.3.6.1.3 and B.3.6.1.4.

NOTE — These last two elements provide necessary functionality to support link protocol analyzers.

B.3.6.4 Address-list-query SPDU

The address-list-query SPDU requests the receiving DLE to search its forwarding database (of DL-addresses recognized for either local reception or forwarding) for DL-addresses which meet the query criteria, starting at the beginning-of-list or continuation-point specified in the query, and to include as many of those DL-addresses which meet the criteria as possible in a address-list-reply SPDU, concluding the SPDU with an end-of-list or continuation-point list element.

Table B.49 — Address-list-query SPDU

octet index	contents of field
1	0101 0VVV
2 **3**	selection criteria
4 **5** **6** **7**	request identifier
8 **…**	parameter-list element

The primary purpose of this SPDU is to provide bridges with a means of building and maintaining their forwarding databases.

The address-list-query SPDU shall be formatted as specified in Table B.49:

a) The SPDU header octet, octet 1, shall be encoded as 0101 0VVV, specifying a address-list-query SPDU.

Table B.50 — DL-address selection criteria

include standard addresses	**reserved**			**DLSEP-address**	**subscriber DLCEP**	**publisher DLCEP**	**peer DLCEP**
A	B (=0)	C (=0)	D (=0)	E	F	G	H
15	14	13	12	11	10	9	8

reserved			**constrained responder DLSAP-address**	**unconstrained responder DLSAP-address**	**initiator DLSAP-address**	**basic DLSAP-address**	**group DL-address**
J (=0)	K (=0)	L (=0)	M	N	P	Q	R
7	6	5	4	3	2	1	0

b) Octets 2 and 3 specify the selection criteria for DL-addresses to be included within a responder's address-list-reply SPDU. They shall be encoded as two octets of individual selection criteria, as specified in Table B.50, with octet 2 consisting of fields A through H, and octet 3 of fields J through R:

1) If field A=1, then the standard DL-addresses specified in Tables A.4, A.5 and A.6 which meet the other selection criteria shall be included in the reply list; otherwise (field A=0) they shall be excluded from the reply list.

2) Fields B, C, D, J, K and L are reserved for future standards use and shall be encoded as zero.

3) If field E=1, then active DLSEP-addresses of the receiving DLE shall be included in the reply list; otherwise (field E=0) they shall be excluded from the reply list.

4) If field F=1, then (publisher) DLCEP-addresses of subscriber DLCEPs within the receiving DLE, or forwarded by the receiving DLE, shall be included in the reply list; otherwise (field F=0) they shall be excluded from the reply list.

5) If field G=1, then DLCEP-addresses of publisher DLCEPs within the receiving DLE, or forwarded by the receiving DLE, shall be included in the reply list; otherwise (field G=0) they shall be excluded from the reply list.

6) If field H=1, then DLCEP-addresses of peer DLCEPs within the receiving DLE, or forwarded by the receiving DLE, shall be included in the reply list; otherwise (field H=0) they shall be excluded from the reply list.

7) If field M=1, then DLSAP-addresses of DLSAPs within the receiving DLE, or forwarded by the receiving DLE, whose DL(SAP)-role is CONSTRAINED-RESPONDER, shall be included in the reply list; otherwise (field M=0) they shall be excluded from the reply list.

8) If field N=1, then DLSAP-addresses of DLSAPs within the receiving DLE, or forwarded by the receiving DLE, whose DL(SAP)-role is UNCONSTRAINED-RESPONDER, shall be included in the reply list; otherwise (field N=0) they shall be excluded from the reply list.

9) If field P=1, then DLSAP-addresses of DLSAPs within the receiving DLE, or forwarded by the receiving DLE, whose DL(SAP)-role is INITIATOR, shall be included in the reply list; otherwise (field P=0) they shall be excluded from the reply list.

10) If field Q=1, then DLSAP-addresses of DLSAPs within the receiving DLE, or forwarded by the receiving DLE, whose DL(SAP)-role is BASIC, shall be included in the reply list; otherwise (field Q=0) they shall be excluded from the reply list.

11) If field R=1, then group DL-addresses of DLSAPs within the receiving DLE, or forwarded by the receiving DLE, shall be included in the reply list; otherwise (field R=0) they shall be excluded from the reply list.

c) Octets 4 through 7 shall encode a non-zero request-identifier provided by the requesting DLE.

NOTE — This identifier enables the requesting DLE to associate a reply with its corresponding request.

d) Octets 8 and following shall encode either one begin/end-of-list element or one continuation-of-list element, as specified in B.3.6.1.1 and B.3.6.1.2. The encoded element shall specify to the receiving DLE where, within its forwarding database, the search should begin:

1) If a begin/end-of-list element is specified, it indicates that the query is interrogating the entire forwarding database of the receiving DLE.

2) If a continuation-of-list element is specified, it indicates that the query is interrogating the forwarding database of the receiving DLE continuing from the specified point. For this reason, the continuation-of-list element shall be identical to that returned by a prior address-list-reply SPDU received from the same correspondent DLE.

B.3.6.5 Address-list-reply SPDU

Table B.51 — Address-list-reply SPDU

octet index	contents of field
1	0101 1VVV
2 **3** **4** **5**	request identifier
6 **…**	parameter-list elements

The address-list-reply SPDU is sent in response to an address-list-query SPDU. Its primary purpose is to provide bridges with a means of building and maintaining their forwarding databases.

The address-list-reply SPDU may be addressed to a DLE on another link. It shall be formatted as specified in Table B.51:

a) The SPDU header octet, octet 1, shall be encoded as 0101 1VVV, specifying an address-list-reply SPDU.

b) Octets 2 through 5 shall encode the non-zero request-identifier provided by the requesting DLE in the address-list-query SPDU to which this SPDU is a reply.

NOTE — This identifier enables the querying DLE to associate the reply with its corresponding query.

c) Octets 6 and following shall encode zero or more SHORT DL-address and LONG DL-address elements, as specified in B.3.6.1.3 and B.3.6.1.4, specifying DL-addresses which meet the selection criteria specified in the corresponding address-list-query SPDU to which this is a reply.

The last element in the parameter-list shall encode either one begin/end-of-list element or one continuation-of-list element, as specified in B.3.6.1.1 and B.3.6.1.2. If the encoded element is a begin/end-of-list element, it shall specify to the querying DLE that the requested search of the replying DLE's forwarding database is complete; if it is a continuation-of-list element, then the querying DLE can request that the search be continued at the point indicated in that continuation-of-list element.

The length of this SPDU may approach 256 octets.

Annex C — (normative) DL-bridge elements of procedure and bridge sub-protocol

(This annex forms an integral part of this standard.)

The ISO/IEC bridge protocol, ISO/IEC 10038, specifies management and operation of the MAC-relay entities of a bridged local area network. This clause specifies additions and modifications to ISO/IEC 10038 which transform it into a bridge protocol suitable for use with Fieldbus DL-relay entities.

In the following text references to clauses of this part of the Fieldbus DL protocol specification are preceded by [FB], references to clauses of ISO/IEC 10038 are preceded by [IL].

C.1 Global changes

Throughout [IL] replace all references to

a) "MAC" *by* "DL" *or* "DL-," *as appropriate*;

b) "Media Access Control" *by* "Data Link;"

c) "Sublayer" *by* "layer;"

d) "LAN" *by* "link;"

e) "Bridged Local Area Network" *by* "extended link;"

f) "user priority" *with* "DLL-priority;"

g) "frame" *with* "DLPDU;"

h) "LLC" *with* "upper DLL;" and

i) "section" *with* "clause."

Throughout [IL] change numeric references to subclauses of [IL] by preceding the reference, if appropriate, with the annex identification "C." to provide a means of discriminating between cross-references within this annex [IL] and references within the annex to the main (non-annex) body of this part [FB] of this international standard.

Make all other style, format and nomenclature changes required by IEC Directives Part 3, and renumber or reletter the items within each subclause as appropriate after the following changes:

C.2 Changes to [IL/1]: Introduction

Modify the following indicated paragraphs of [IL/1.5] as shown:

(1) delete paragraph (b)

(4) delete this paragraph

(7) *replace* "may" *with* "shall"

(9) *replace paragraph with* "shall use 32-bit DL-addresses and their 48-bit MAC-address representations as specified in [IL/3.1.2] and described in [FB/A.4].

(11) *replace* "may" *with* "shall" *and delete* "optional"

(12) *delete paragraphs (a) and (b)*

(16) *replace* “may” *with* “shall”

(17) *replace* “may” *with* “shall”

(19) *replace* “may” *with* “shall”

(20) *replace paragraph with* “shall support remote management, as specified in [IL/7] and in Part 7 of this International Standard”

Delete [IL/1.6 and 1.7].

C.3 Changes to [IL/2]: Support of the DL-Service

Replace the first paragraph of [IL/2.1] with “The DL-service provided to end stations attached to an extended link is supported by the bridges in that extended link. Bridges support all of the DL-services which involve multi-end-station communication or scheduling.”

Delete “Destination” *from [IL/2.2(1)].*

Replace paragraph [IL/2.2(3)] with “the DL-addresses of end-stations are not restricted by the topology and configuration of the extended link, except as specified in [FB/A.2.1] for link-designator values between 1000_{16} and $FEFF_{16}$, inclusive.”

Delete paragraph [IL/2.3.2(2c)].

Replace paragraph [IL/2.3.3] with “The operation of Bridges is unlikely to misorder DLPDUs transmitted with the same DL-priority. Such misordering can occur

a) under conditions where DL-redundancy is employed to provide multiple simultaneously-usable paths between the two end-stations; or

b) immediately subsequent to reconfiguration of the DL-path between the two end stations.

NOTE — Such reconfiguration can occur only as the result of DL-management action or during recovery from bridge or Ph-layer entity failures.

Replace the second paragraph of [IL/2.3.7] with “For DLPDUs relayed between links, the bridge shall modify the FC octet of each DLPDU and make compensatory modifications to the FCS of that DLPDU as specified in [FB/6.1.1.3].”

Insert after [IL/2.3.10] “**2.3.11 Time synchronization**. The bridges establish and preserve the shared sense of DL-time throughout the extended link.”

Replace [IL/2.4] with “**2.4 Forwarding of DLPDUs**.

“A bridge receiving a DLPDU examines the DLPDU’s frame type and first address, from which the bridge determines whether and to where the DLPDU should be forwarded.

“A bridge forwards a CA, CD, DC, DT, EC, ED or RC DLPDU, together with its received FCS, toward those links which have DLEs which should receive the DLPDU’s first DL-address. No other forwarding of DLPDUs occurs.

“If the received DLPDU is a CA, CD or ED DLPDU, then the bridge replies immediately with a SR DLPDU on the link from which the CA, CD or ED DLPDU was received.”

Delete [IL/2.5].

C.4 Changes to [IL/3]: Principles of operation

Insert before [IL/3.1 (2)]:

“(2)Relay of the shared sense of DL-time.”

Modify the indicated paragraphs of [IL/3.1.1] as shown:

(3) *Replace with* “Frame discard if the DLPDU type is not CA, CD, DC, DT, EC, ED, RC or SR; and frame discard if the DLPDU is a detected duplicate of a prior CA, CD or ED DLPDU, presumably caused by loss of a prior immediate retry (see 7.4.4, 7.5.4 and 7.6.4).”

(3+)*Insert before (4)* “Possible frame transformation or discard if the DLPDU type is DT or SR (see 7.7.4.3(b) and 7.8.4.3(a)).”

(5) Delete this paragraph.

(8) Delete this paragraph.

(9) *Replace with* “modification of a frame control octet, with compensatory modification of the DLPDU’s FCS.”

(9+)*Insert before (10)* “FCS calculation on transformed DT or SR DLPDUs (both of which become DT DLPDUs).”

Append to [IL/3.1.2 (3)] “… and queries to other bridges and end-stations on the local link.”

Replace [IL/3.2] with “**3.2 Bridge architecture**. Each bridge port receives and transmits DLPDUs to and from the link to which it is attached. The DL-relay-entity handles the functions of relaying DLPDUs between bridge ports, filtering DLPDUs, and learning filtering information.

“The bridge protocol entity handles calculation and configuration of the extended link topology. The bridge protocol entity and other higher-layer protocol users, such as DL-management, make use of higher-level DLL services, which are provided in a port-independent manner.

“Figures 3-2 and 3-3 of [IL] illustrate a bridge and its ports, and the architecture of the bridge for a bridge with two ports. A bridge may have more than two ports. Where Fractional Duty Cycle (FDC) nodes exist on a link, it is also meaningful for a bridge to have just a single port, since a one-port bridge can supply the DLPDU store-and-forward services needed to permit communication with or among FDC nodes.”

Delete the first paragraph of [IL/3.3].

Replace Figures 3-2 and 3.3 of [IL] with Figures C.1 and C.2 to replace the term “LAN” *with* “local link” *and to delete the terms* “LLC entities” *and* “MAC service interfaces:”

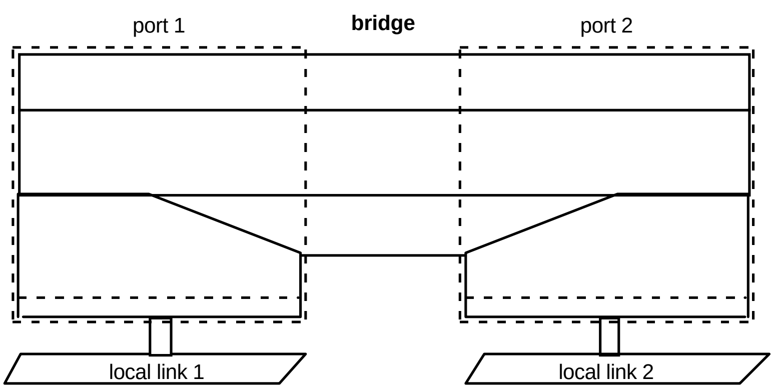


Figure C.1 — Replacement for [IL] Fig 3-2 Bridge ports

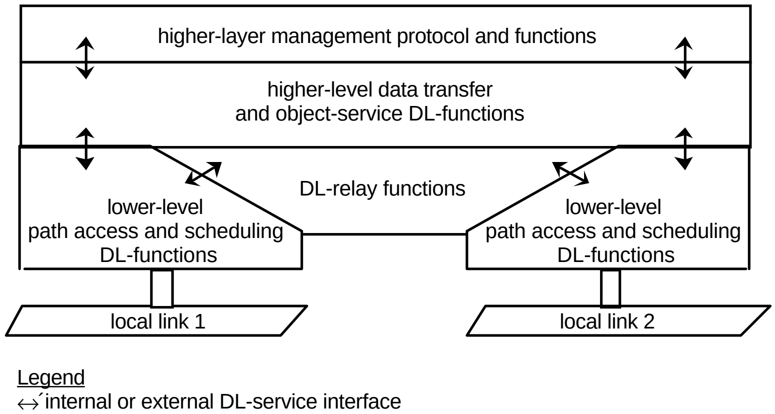


Figure C.2 — Replacement for [IL] Fig 3-3 Bridge architecture

Delete the paragraph of [IL/3.3] which begins “Each Bridge Port shall support ….”

Replace the title of Figure 3.4 of [IL] with “**Relaying DLPDUs**.”

Replace the title of Figure 3.5 of [IL] with “**Observation of link traffic**.”

Delete the dotted blocks “LLC” *from Figure 3-6 of [IL].*

Replace [IL/3.5] with “**3.5 Frame reception**. The lower-level DLL entity associated with each bridge port examines all DLPDUs received from the link to which it is attached. Frames that are detected to be in error are discarded; this includes frames whose FCS is in error. Frames that are detected to be duplicates of prior received frames, due to immediate retry by the current holder of a scheduled or delegated token, are also discarded after generation of the required immediate reply. All other frames shall be submitted to the learning process.

“Received DC, EC and RC DLPDUs [IL/2.4] shall be submitted to the forwarding process. Received CA, CD and ED DLPDUs [IL/2.4] shall be submitted to the forwarding process except when they are discarded as duplicates of prior received DLPDUs. Received DT DLPDUs [IL/2.4] containing at least one explicit DL-address shall be submitted to the forwarding process.

“Other DLPDU types shall not be relayed by the bridge. However, the bridge may be required to transform a received SR DLPDU, or a received DT DLPDU without explicit DL-addresses and with null user data, to a DT DLPDU with one explicit DL-address and null user data, and forward that transformed DLPDU as if it had just been received (see 7.7.4.3(b) and 7.8.4.3(a)).

“DLPDUs addressed to the bridge as an end station shall be submitted to upper-DLL processing except when they are discarded as duplicates of prior received DLPDUs.”

“Locally-originated DLPDUs shall be submitted directly to the forwarding process by the upper-DLL functions.”

Replace [IL/3.6] with “**3.6 Frame transmission**. The lower-level DLL entity associated with each bridge port transmits DLPDUs submitted to it by higher DLL functions. Relayed, transformed and locally-originated DLPDUs are submitted for transmission by the forwarding process.”

Replace [IL/3.7.1 (3) (a)] with "The DLPDU specifies a destination DL-address and the Filtering Database indicates that DLPDUs with this value of the destination address should be forwarded through the transmission Port (as would happen, for example, if the destination address specified a local link accessible through that transmission Port)."

Replace [IL/3.7.1 (3) (b)] with "The DLPDU specifies only a source DL-address and the Filtering Database indicates that source-only DLPDUs with this value of the source address should be forwarded through the transmission Port (as would happen, for example, if expected receivers of DLPDUs with that source address were attached to a local link accessible through that transmission Port)."

Delete [IL/3.7.2].

Delete [IL/3.7.4].

Replace [IL/3.7.5] with "3.7.5 Modification of DLPDUs. When forwarding a DLPDU, it is sometimes necessary for a bridge to alter the final-use subfield of a DLPDU's frame control field. Specifically, the bridge may have to complement the value of the final-use subfield of the frame control field. This modification, with a corresponding FCS modification, shall be performed as specified in [FB/6.1.1.3]."

Delete Tables 3-1 and 3-2 of [IL].

Replace the first paragraph of [IL/3.8] with "**3.8 The learning process**. The Learning Process observes the source DL-addresses, and in some cases the destination DL-addresses and CE-parameters, of DLPDUs received on each Port and updates the Filtering Database conditionally on the state of the receiving Port.

Replace the third paragraph of [IL/3.8] and its sub-paragraphs with "The Learning Process may deduce the path through the extended link to particular end stations, by inspection of the source DL-address fields of CA, CD, DC, DT, ED and RC DLPDUs, and by inspection of the EC-parameters and all of the DL-address fields of received EC DLPDUs.

"(1) For the source DL-address of a CA, CD, DC, DT, ED or RC DLPDU, and for each of the two source DL-addresses of an EC DLPDU, if

"(a) the link-designator component of that DL-address is in the range 1000_{16} to V(ML), inclusive, where V(ML) is as defined in [FB/5.7.6.1];

"(b) the Port from which the DLPDU was received was in a state that allows learning [IL/4.4];

"(c) a static entry [IL/3.9, 3.9.1] for the link-designator component of that source DL-address does not already exist; and

"(d) the resulting number of entries would not exceed the capacity of the Filtering Database;

"then the Learning Process shall create or update a dynamic entry [IL/3.9, 3.9.2] in the Filtering Database for the link-designator component of that source DL-address, associating the Port on which the DLPDU was received with that link-designator component.

"(2) For the source DL-address of a CA, CD, DC, DT, ED or RC DLPDU, and for each of the two source DL-addresses of an EC DLPDU, if

"(a) the link-designator component [FB/Annex A/1] of the source DL-address is in the range 0080_{16} to $0FFF_{16}$, inclusive;

"(b) the Port from which the DLPDU was received was in a state that allows learning [IL/4.4];

"(c) a static entry [IL/3.9, 3.9.1] for that source DL-address does not already exist; and

"(d) the resulting number of entries would not exceed the capacity of the Filtering Database;

"then the Learning Process shall create or update a dynamic entry [IL/3.9, 3.9.2] in the Filtering Database for that source DL-address, associating the Port on which the DLPDU was received with the DL-address.

"(3)For an EC DLPDU, if

"(a) the sender's-DLCEP-class parameter of the DLPDU's EC-parameters specifies PUBLISHER [FB/8.1(a.4)];

"(b) the Port from which the DLPDU was received was in a state that allows learning [IL/4.4];

"(c) a static entry [IL/3.9, 3.9.1] for the DLPDU's first source DL-address does not already exist; and

"(d) the resulting number of entries would not exceed the capacity of the Filtering Database;

"then the Learning Process shall create or update a dynamic Publisher entry [IL/3.9, 3.9.2] in the Filtering Database for that source DL-address, associating the Ports to which the DLPDU will be forwarded to the subscriber ports of that DL-address.

"(4) For an EC DLPDU, if

"(a) the sender's-DLCEP-class parameter of the DLPDU's EC-parameters specifies SUBSCRIBER [FB/8.1(a.4)];

"(b) the Port from which the DLPDU was received was in a state that allows learning [IL/4.4];

"(c) a static entry [IL/3.9, 3.9.1] for the DLPDU's destination DL-address does not already exist; and

"(d) the resulting number of entries would not exceed the capacity of the Filtering Database;

"then the Learning Process shall create or update a dynamic Publisher entry [IL/3.9, 3.9.2] in the Filtering Database for that destination DL-address, associating the Port on which the DLPDU was received as one of the subscriber ports of that DL-address.

"The Learning Process may also query other DLEs on the local link by sending DL-address list query SPDUs [FB/Annex B] to the DL-support functions of those DLEs and using the resultant DL-address list reply SPDUs [FB/Annex B] as a basis for adding or removing entries from the Filtering Database."

Replace the second to last paragraph of [IL/3.8] with "If the Filtering Database is already filled up to its capacity, but a new entry would otherwise be made, then DL-management shall be informed of the Filtering Database overflow, and the new entry shall not be made."

Replace the last sentence of the first paragraph of [IL/3.9] with "It supports queries by the Forwarding Process as to whether DLPDUs with given values of the destination DL-address, or of the source DL-address in DLPDUs specifying only an explicit source DL-address, should be forwarded to a given Port."

Replace the second paragraph of [IL/3.9] with "The Filtering Database shall be capable of containing both static entries [IL/3.9.1] and dynamic entries [IL/3.9.2], for both full DL-addresses and DL-address link-designator components [FB/Annex A/1]).

"Static and dynamic entries shall not both exist for the same full DL-address; a dynamic entry shall not be created if a corresponding static entry for the same full DL-address already exists; and creation of a static entry shall cause the removal of a corresponding dynamic entry for the same full DL-address if one exists.

"Static and dynamic entries shall not both exist for the same DL-address link-designator component; a dynamic entry shall not be created if a corresponding static entry for the same partial DL-address already exists; and creation of a static entry shall cause the removal of a corresponding dynamic entry for the same partial DL-address if one exists.

“Each entry in the Filtering Database shall be of one of five classes:

“(a) A DL-address link-designator component entry, which specifies the default single port to which a DLPDU with a destination DL-address containing this link-designator component should be forwarded when the Filtering Database does not contain a specific entry for the destination DL-address specified in the DLPDU.

“(b) A group DL-address entry, which specifies, separately for each receiving port, the set of ports to which a DLPDU whose destination DL-address equals the specified DL-address should be forwarded.

“(c) DLSAP-address entry, which specifies the single port to which a DLPDU whose destination DL-address equals the specified DL-address should be forwarded.

“(d) A peer DLCEP-address entry, which specifies the single port to which a DLPDU whose destination DL-address equals the specified DL-address should be forwarded.

“(e) A publisher DLCEP-address entry, which specifies

“(i) the single port to which a CA, CD, DT, EC or ED DLPDU whose destination DL-address equals the specified DL-address should be forwarded; and

“(ii) the set of ports, all different from (i), to which a DC, DT, EC or RC DLPDU which contains no destination DL-address, and which specifies this DL-address as an explicit source DL-address, should be forwarded.”

Replace the second and following paragraphs of [IL/3.9.1] with “A static entry may specify either a full DL-address or a link-designator component of a DL-address.”

Replace the last five paragraphs of [IL/3.9.2], beginning “Dynamic entries specify …”, *with* “A dynamic entry may specify any full DL-address or DL-address link-designator component.”

Replace all of [IL/3.12] with “**3.12 Addressing**.” *That is, delete all text following the title up to but not including [IL/3.12.1].*

In the second paragraph of [IL/3.12.3], replace “LLC Entities” *with* “DLEs,” *and replace* “Standard LSAP … users of LLC” *with* “DL-bridge functions of DLEs (see annex A).”

Replace the third paragraph of [IL/3.12.3] with “Each DL-UNITDATA request primitive causes the transmission of a DLPDU which conveys the BPDU in its user-data field as a complete DLSDU.”

Delete the fourth paragraph of [IL/3.12.3].

In the fifth paragraph of [IL/3.12.3], delete “, within the scope of the LSAP Assignment.”

Delete the last sentence of the seventh paragraph of [IL/3.12.3], which states “This group address … are transmitted.”

Delete the eighth paragraph of [IL/3.12.3].

Replace the last paragraph of [IL/3.12.4] with “[FB/Annex A] specifies standard DL-addresses for the DL-bridge functions of bridge DLEs.”

Delete the second and third paragraphs of [IL/3.12.5].

Delete [IL/3.12.6] and Tables 3-4, 3-5, 3-6 and 3-7 of [IL].

C.5 Changes to [IL/ 4]: The spanning tree algorithm and protocol

In the second paragraph of [IL/4.3.4], change “must” *to* “should.”

In Tables 4-2 and 4-3, replace the first line after each table with “All times are in seconds when the local link data rate is 1 Mb/s or greater; otherwise all times are in minutes.”

In Table 4-3, replace the line

Bridge Forward Delay	15.0	—	4.0 – 30.0”

with

Bridge Forward Delay	0.0	—	0.0 – 30.0”

Insert before [IL/4.6.11] “**4.6.10.3.5** The port’s DLE shall attempt to become the LAS for the local link to which it is attached.”

C.6 Changes to [IL/5]: Encoding of bridge protocol data units

Replace the third paragraph of [IL/5.2.5] with “The third most significant octet is assigned the value of the initial octet of the 48-bit Bridge Address. The fourth through eighth octets are similarly assigned the values of the second to the sixth octets of the 48-bit Bridge Address, respectively.”

Delete the fourth paragraph of [IL/5.2.5].

C.7 Changes to clause 6: Bridge management

Replace [IL/6.3(5)] with “(5) IA5 String, for all text strings.”

Replace [IL/6.7.5.1.2 (2)] with “(2) Address — full DL-address or DL-address link-designator component of the entry.”

Replace [IL/6.7.5.1.2 (3)] with “(3) Entry class; associated port or port maps:

> “(a) DL-address link-designator component — the default single port to which a DLPDU with a destination DL-address containing this link-designator component should be forwarded when the Filtering Database does not contain a specific entry for the destination DL-address specified in the DLPDU
>
> “(b) group-DL-address — an array which specifies, separately for each receiving port, the set of ports to which a DLPDU specifying this DL-address as a destination DL-address should be forwarded.
>
> “(c) DLSAP-address — the single port to which a DLPDU specifying this DL-address as a destination DL-address should be forwarded.
>
> “(d) peer DLCEP-address — the single port to which a DLPDU specifying this DL-address as a destination DL-address should be forwarded.
>
> “(e) publisher-DLCEP-address —
>
> > “(i) the single port to which a DLPDU specifying this DL-address as a destination DL-address should be forwarded; and

“(ii) the set of ports, all different from (i), to which a source-only DC, DT, EC or RC DLPDU specifying this DL-address as an explicit source DL-address should be forwarded.”

Replace [IL/6.7.5.2.2 (2)] with “(2)Address — full DL-address or DL-address link-designator component of the desired entry.”

Replace [/IL/6.7.5.3.2 (2)] with “(2)Address — full DL-address or DL-address link-designator component of the desired entry.”

Replace [/IL/6.7.5.3.3 (1)] with “(1)Address — full DL-address or DL-address link-designator component of the desired entry.”

Replace [IL/6.7.5.3.3 (3)] with “(3)Entry class; associated port or port maps; as specified in 6.7.5.1.2 (3).”

Replace [IL/6.7.5.4.3 (3) (a)] with “(a)Address — full DL-address or DL-address link-designator component of the desired entry.”

Replace [IL/6.7.5.4.3 (3) (c)] with “(c)Entry class; associated port or port maps; as specified in [IL/6.7.5.1.2 (3)].”

C.8 Changes to [IL/7]: Management protocol

Replace the entire clause after the title with:

“The Fieldbus management protocol is specified in Part 7 of this International Standard. Additional DL-Management capabilities are specified in Annexes B and D of this part of this international standard.”

C.9 Changes to [IL Annex A]: PICS proforma

Delete items 1a, 1b and 2a.

Reword item 2c to read “Are CA, CD, DC, DT, EC, ED, RC and SR DLPDUs the only types of DLPDU relayed, possibly after transformation as specified in 7.7.4.3(b) and 7.8.4.3(a)?”

Delete items 2d, 2f, 2g, 2n, 2o, 2q, 2r, 2s, 4, 4a, 4b, and 5b.

Reword item 5g to read “Does each static or dynamic entry specify either a full DL-address or the link-designator component of a DL-address?”

Reword item 5i to read “Does each entry which specifies a link-designator component of a DL-address specify a port number?”

Renumber old item 5j to 5p.

Add new item 5j “Does each entry which specifies a group-DL-address specify an array of sets of ports, with one set for each receiving port?”

Add new item 5k “Does each entry which specifies a DLSAP-address specify a port number?”

Add new item 5m “Does each entry which specifies a peer DLCEP-address specify a port number?”

Add item 5n “Does each entry which specifies a publisher-DLCEP-address specify

“a) a port number; and

“b) separately, a set of ports?”

Reword item 7d to read “Can static entries be made for link-designators in the range 1000_{16} to $FEFF_{16}$?”

Replace item 7f with “Can static entries be made for DLSAP- addresses?”

Renumber old item 7g to 7j.

Add new item 7g “Can static entries be made for peer DLCEP-addresses?”

Add new item 7h “Can static entries be made for publisher DLCEP-addresses?”

Delete items 9a, 9b, 9c, 10b, 10e, 10f, 11a, 11b, 11d, 11e, 11f, and 12a.

Replace item 10c with “Are all BPDUs addressed to the DL-bridge functions of bridge DLEs?”

C.10 Delete [IL/Annex C] and [IL/Annex D], both of which relate to source-routing

Annex D — (normative) DL-management-information

(This annex forms an integral part of this standard.)

D.1 Scope

This clause enumerates the set of DL-parameters, defined as variables in 5.7 of this part of this International Standard, which need to be preconfigured before proper DLE operation is possible.

Full DL-management is specified in Part 7 of this International Standard.

D.2 DLE configuration parameters

The following parameters, defined as variables in 5.7, are required to support the three classes of DLE operation.

D.2.1 Node-specific DL-configuration parameters

The following parameters shall be assigned the appropriate values for the DLE before the DLE is permitted to transmit any DLPDU on the local link:

V(TN) this-node, defined in 5.7.1.8.

D.2.2 Additional node-independent DL-configuration parameters

The following DL-parameters shall be assigned the then-current values for the local link before the DLE is permitted to transmit any DLPDU on the local link:

NOTE 1 – Each PN DLPDU conveys the values of the following parameters, plus those of critical PhL parameters. Thus a DLE which receives a PN DLPDU is able to configure itself, and its associated PhE, so that the DLE and associated PhE can reply to that or a subsequent PN DLPDU, starting the process of integrating the replying DLE into an already-functioning network.

V(ST) slot-time, defined in 5.7.1.1;

V(MRD) maximum-response-delay, defined in 5.7.1.3; and

V(MID) minimum-inter-DLPDU-delay, defined in 5.7.1.12.

The following DL-parameters shall be assigned the then-current values for the local link before the DLE is permitted to transmit any DLPDU other than a PR DLPDU on the local link:

NOTE 2 – A single PR DLPDU is able to convey a probe-response SPDU to the local LAS, which will cause the current local-link values of the following parameters to be returned in a DL-node-activation SPDU (see Annex B).

V(PhLO) per-DLPDU-PhL-overhead, defined in 5.7.1.2;

V(MRC) maximum-retry-count (see 5.7.1.5);

V(NRC) network-repeat-count (see 5.7.1.6);

V(NDL) network-DLPDU-lifetime (see 5.7.1.7);

V(TL) this-link, defined in 5.7.1.9;

V(MEP) MAC-embedding-prefix, defined in 5.7.1.10; and

V(TSC) time-synchronization-class, defined in 5.7.1.25.

D.2.3 Additional node-independent DL-configuration parameters for link-master class DLEs

The following DL-parameters shall be assigned the then-current values for the local link before the DLE is permitted to function as a DLE of the Link-master (see 5.6.2) or Bridge (see 5.6.3) class (that is, as a potential LAS ready to assume the LAS role), and transmit on the local link:

—all of the parameters defined in D.2.2.

V(MST) maximum-scheduled-traffic, defined in 5.7.5.5;

V(DMDT) default-minimum-token-delegation-time, defined in 5.7.5.7;

V(DTHT) default-token-holding-time, defined in 5.7.5.8;

V(LTHT) link-maintenance-token-holding-time, defined in 5.7.5.9;

V(TTRT) target-token-rotation-time, defined in 5.7.5.11;

V(FUN) first-unpolled-node id, defined in 5.7.5.15;

V(NUN) number-of-consecutive-unpolled-nodes ids, defined in 5.7.5.16;

V(MTHA) maximum-token-holding-time-array, defined in 5.7.5.10;

V(TDP) time-distribution-period, defined in 5.7.5.18;

V(MICD) maximum-inactivity-to-claim-LAS-delay, defined in 5.7.5.19; and

V(LDDP) LAS-data-base-distribution-period, defined in 5.7.5.20.

D.2.4 Additional node-independent DL-configuration parameters for bridge class DLEs

The following DL-parameters shall be assigned the then-current values for the local link before the DLE is permitted to function as a DLE of the Bridge (see 5.6.3) class and transmit on the local link:

—all of the parameters defined in D.2.2;

—all of the additional parameters defined in D.2.3;

V(ML) maximum-link, defined in 5.7.6.1; and

—all of the configuration parameters defined in ISO/IEC 10038 part 6.

D.2.5 Node-independent Ph-configuration parameters required for minimal DL-communication

This sub-clause enumerates the minimal set of Ph-parameters, defined in 6.2 and 8.4 of Part 2 of this International Standard, which need to be preconfigured before proper PhE operation is possible.

NOTE 1 – These Ph-parameters, defined in table 1 of Part 2 of this International Standard, are included to make it possible for a device, newly connected to a Fieldbus local link, to learn the minimal information required for successful communication with that local link's LAS, and thereby request any additional needed PhL and DLL configuration information, after which remote management becomes possible.

V(PhGE) post-transmission-gap-extension-units;

NOTE 2 – Although this measure is named “post-transmission-gap -extension” in Part 2 of this International Standard, it is a measure of the amount of observable non-transmission required between any two transmissions on the local medium, as observed from any point on that medium, whether the two transmissions are from the same source PhE or from two different source PhEs. Thus it could just as correctly be named “pre-transmission-gap-extension,” or even more correctly “inter-transmission-gap-extension.”

V(PhPE) preamble-extension-units; and

V(PhIS) maximum inter-channel signal skew.

D.3 DLE-collected fault-management data

NOTE – DLE-collected fault-management data are designed to provide the maximum amount of information for network and end-system fault management while imposing the least performance burden on real implementations.

D.3.1 Required statistical measures

Each DLE needs to keep a minimum set of statistical measures to facilitate the identification of communications errors which render the local link inoperable, and the sources of those errors, where that information is available. These statistical measures provide a partial basis for network fault management.

NOTE 1 – These statistical measures may be read through higher-layer network-management protocols.

In general, these statistical measures take the form of event counters, where a network transmission or reception event results in the incrementation of a counter. Conceptually, these counters are of unbounded length, are reset only on DLE activation or DLE reset, and record the number of events of the appropriate type which have occurred since the DLE was last activated or reset.

In general, counters which record error events need only be two octets in length, while counters that record events expected during normal error-free operation should be longer to permit infrequent loss-free readout by remote DL-management. The remainder of this subclause specifies both the minimum required size, and, where different, the recommended size, of each counter.

These counters shall be reset only upon DLE activation or DLE reset; they shall not be reset by any other local or remote means.

NOTE 2 — This restriction on resetting the counters is necessary if they are to be meaningful in a multi-manager environment.

D.3.1.1 Transmission-related statistical measures

a) The total number of jabber (streaming) faults which were detected during transmission, if such faults are detectable separately from other modem or MAU faults, or zero if such faults are not separable. This counter shall be at least two octets in length.

b) The total number of DLPDUs which were transmitted as immediate retries of the immediately-prior transaction (see 3.3.18). This includes DLPDUs retransmitted by the DLE when it is functioning in an initiator role, and DLPDUs retransmitted by the DLE when it is functioning in a responder role. This counter shall be at least two octets in length.

c) The total number of DLPDUs which were transmitted by the DLE, but excluding and not counting those DLPDUs counted in category (b). This counter shall be at least two octets in length, and should be at least four octets in length.

D.3.1.2 Reception-related statistical measures

The following two statistical measures shall be kept

– separately for each PhL-medium attached to the DLE, if an implementation so permits; or

– aggregated for all of the PhL-media attached to the DLE.

NOTE 1 – These statistical measures facilitate identification of PhE-related problems.

a) The total number of modem-internal or MAU-internal faults, which were detected during transmission or reception, plus the total number of jabber (streaming) faults not reported in category D.3.1.1(a). This counter shall be at least two octets in length.

NOTE – Implementations which do not distinguish jabber (streaming) faults from modem-internal or MAU-internal faults will count all such faults in this category.

b) The total number of **END-OF-ACTIVITY** indication events (see 5.4.1.2) at the PhE interface for which there was not an immediately preceding **END-OF-DATA** indication event. This counter shall be at least two octets in length.

NOTE – This is the number of DLPDUs and noise intervals for which reception commenced, but in which a partially-received DLPDU was discarded due to detected framing or coding errors.

Implementation NOTE — The above events at the virtual DLE-PhE interface may not have an exact equivalence in real implementations. In such a case implementors should use available information to approximate as closely as possible the apparent number of messages for which reception commenced and did not result in receiving a complete well-formed frame. If the number of FCS errors cannot be segregated for separate accumulation as in (d), then it should be aggregated within this accumulation (but it should not be counted twice).

When a DLE is implemented so that it does not actually receive transmitted DLPDUs, but processes them as if they had been received, then the following counters shall not be incremented as a result of a transmission by the DLE.

NOTE 2 – This is equivalent to saying that the DLE operates in a physically half-duplex mode.

The following additional statistical measures shall be kept for each DLE, where that information is available:

NOTE 3 – These statistical measures facilitate identification of link-related or path-related, but not PhL-media-related, problems.

c) The total number of partially-received DLPDUs which were discarded after examination of the DLPDU’s frame control field and first address subfield, if any, and for which the associated FCS was not checked for correctness. This counter shall be at least two octets in length, and should be at least four octets in length.

NOTE – If the receiving DLE takes any action based on the received DLPDU, such as an LAS DLE checking the frame control octet’s final-use subfield, or any DLE noting the destination address of a CA, CD or ED DLPDU to enable reception of the reply DLPDU, then the receiving DLE necessarily has to check the correctness of the FCS before acting on the information in the DLPDU. In such a case this specific statistical measure cannot, and does not, apply.

d) The total number of completely-received DLPDUs which were discarded due to a detected incorrect FCS (see 6.1.1.2) and which were not counted in (b). This counter shall be at least two octets in length.

e) The total number of DLPDUs which were discarded as inferred duplicates due to immediate retry of an immediately-prior transaction. This counter shall be at least two octets in length.

f) The total number of received DLPDUs, with a correct FCS, which were not counted in categories (a) through (e). This counter shall be at least two octets in length, and should be at least four octets in length.

g) The total number of occurrences which the DLE has detected, of a timeout of one of the DLE’s timers which is prescaled by the local link’s slot-time, V(ST). This includes

1) timeouts as initiator of a two-phase transaction requiring an immediate reply;

2) timeouts as LAS while monitoring a delegated token holder;

3) timeouts as a prior LAS while monitoring a transfer of the LAS role to another DLE; and

4) timeouts resulting in sending a CL DLPDU in an attempt to claim the LAS role and (re)-initialize the local link.

This counter shall be at least two octets in length.

D.3.1.3 Additional reception-related statistical measures required of a bridge DLE

A DLE functioning as a port of a bridge shall keep the following statistical measures in addition to those specified in D.3.1.2, where that information is available:

a) The total number of DLPDUs which were stored and forwarded for retransmission by the bridge. This counter shall be at least two octets in length, and should be at least four octets in length.

b) The total number of octets of the DLPDUs specified in (a), not counting FCS octets. This counter shall be at least two octets in length, and should be at least four octets in length.

D.3.2 Additional required DLE-collected fault-management data

a) For each of the last four timeout events counted in D.3.1.3(g), the DLE shall retain the frame control octet and address subfields, if any, of the DLPDU which the DLE last transmitted or received (as appropriate) immediately before the occurrence of that event, where that information is available.

D.3.3 Additional statistical measures

A DLE may also keep additional statistical measures on other aspects of DLE operation to facilitate network and end-system fault management and performance monitoring, where that information is available.

Annex E[4] — (normative) PICS proforma

(This annex forms an integral part of this standard.)

Introduction

To evaluate conformance to a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given OSI protocol. Such a statement is called a Protocol Implementation Conformance Statement (PICS).

E.1 Scope

This annex to Part 4 of this International Standard provides the PICS proforma for the Fieldbus Data Link Layer protocol as specified in the body and other annexes of Part 4 of this International Standard, in compliance with the relevant requirements, and in accordance with the relevant guidelines, given in ISO/IEC 9646-2.

When the supplier completes this proforma, this document becomes the PICS for the specified implementation. The PICS is then used to perform static conformance review of the implementation to determine that all mandatory features are implemented and that all conditional features are correctly supported dependent on the options claimed to be implemented. The PICS also is used as an aid in test case selection.

NOTE — It is anticipated that this PICS will be extended in future revisions of this Annex.

E.1.1 Audience

The PICS proforma shall be used by the supplier of an implementation to state the capabilities of the implementation. The completed PICS shall be used by the test operator and the test analyst to determine the static conformance of an implementation under test (IUT), to select test cases and test suites to use when testing the IUT, and to aid in test results analysis. The completed PICS may be used by a user to determine the adequacy of an implementation.

E.2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this annex to Part 4 of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 9646-1 : 1991, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 1: General concepts.*

ISO/IEC 9646-2 : 1991, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 2: Abstract test suite specification.*

---

4 Copyright release for PICS proformas.

Users of this International Standard may freely reproduce the PICS proforma in thix annex so that it can be used for its intended purpose and may further publish the completed PICS.

E.3 Definitions

This annex to Part 4 of this International Standard uses the following terms defined in ISO/IEC 9646-1:

a) PICS proforma;

b) protocol implementation conformance statement (PICS); and

c) static conformance review.

E.4 Abbreviations

The following abbreviations are used in defining the status of a feature, timer, parameter, or other capability:

M : mandatory;

O : optional;

X : prohibited;

N/A : not applicable; and

C : conditional.

The following abbreviations are used in defining the support of a feature, timer, parameter, or other capability:

Y or **y** or **yes**: implemented; and

N or **n** or **no**: not implemented

The following abbreviations are used in defining the range or existence of a feature, parameter, or other capability:

F : false; and

T : true

References of the form “/n” or “/n.x” refer to the conformance claimed for row “n,” or for row “n” column “x” respectively, in the same table. References of the form “m/n” or “m/n.x” are a corresponding reference to the conformance claimed in table “m.”

E.5 Conformance

The supplier of a protocol implementation which is claimed to conform to Part 4 of this International Standard is required to complete a copy of the PICS proforma provided in this annex and is required to provide the information necessary to identify both the supplier and the implementation.

E.6 Instructions

The supplier of a Fieldbus Data Link protocol implementation shall take the following steps to complete the PICS proforma.

The supplier shall complete the identification section.

The supplier shall enter either a ‘Y’ or an ‘N’ (or equivalents), as defined in E.4, in each entry in the support columns of the following tables. The supplier shall enter a ‘Y’ if the feature, timer, parameter, or capability is implemented. When a conformance requirement defines the allowable or required range or set of values for a capability or parameter, the “allowed” or “required” column contains the specified value(s), and the supplier of the implementation shall only enter a ‘Y’ in the support entry if the implementation completely supports the specified range or set of values. The supplier shall enter a ‘Y’ in the support entry even if access to the feature is not provided as an option (that is, the feature is unconditionally present).

The supplier shall enter an ‘N’ if the feature, timer, parameter, or capability is not implemented at all. If the supplier enters an ‘N’ for a mandatory feature, then the supplier shall attach an explanation as to why the feature was not implemented.

The supplier shall enter the specified ranges, values, default value (as set by the supplier of the implementation) and units of measure, where applicable, for all implemented parameters. Continuous ranges shall be entered in the form: lowest-value ‘:’ highest-value. Series of discrete values or subranges shall be entered from lowest value to highest value, with each value or subrange separated by commas or semicolons.

If the units of measure, or any other column, is not applicable, then a ‘—’ should be recorded in the appropriate entry.

E.7 Identification

Date: ____________________

Data Link Layer Implementation Information

Implementation Name: ____

Implementation Identifier (including version/release):

Vendor Information

Company Name: ____

Technical Representative: ____

Address: ____

____

____

____

Telephone: ____

Facsimile: ____

Telex: ____

E-mail ____

Protocol Identification

Conformance is hereby claimed to IEC 1158-4 : 199x, Digital data communications for measurement and control — Fieldbus for use in industrial control systems —
Part 4 : Data Link Protocol Specification

The protocol versions supported are:____

Overall Conformance Statement

Are all mandatory capabilities of the claimed version(s) of the above protocol fully supported? _________

Are all mandatory capabilities of the profiles(s) claimed under E.8 fully supported? ___________________

Answering "No" to either of these questions indicates non-conformance to the protocol specification. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conforming.

E.8 Implementation profile

This clause identifies the major implementation choices which limit the DL-services and DL-service features which the implementation provides and supports. It is based on clause 11 of this Part of this International Standard.

E.8.1 DL-address, queue and buffer management services

The implementation may, but need not, support LONG DL-addresses (see 11.1).

NOTE — Direct communication and direct management through bridges is based upon the use of LONG DL-addresses.

Supported DL-address lengths				
Item number	**Supported DL-address lengths**	**A) Reference**	**B) Status**	**C) Support**
1	LONG DL-addresses	11.1	O	
2	SHORT DL-addresses	11.1	M	
3	VERY-SHORT DL-addresses	11.1	M	

E.8.2 Data delivery services

E.8.2.1 Data delivery priorities

The implementation must support the TIME-AVAILABLE priority of data transfer. It may also support other priorities: NORMAL and URGENT.

Supported data delivery priorities				
Item number	**Supported data delivery priorities**	**A) Reference**	**B) Status**	**C) Support**
1	TIME-AVAILABLE	11.2 a	M	
2	NORMAL	11.2 a	O	
3	URGENT	11.2 a	C1	
C1 : **if** /2 **then** M **else** X.				

E.8.2.2 Connection-mode data delivery features

The implementation must support UNORDERED peer-to-peer and multi-peer DL-connections. It may also support other DLC data delivery features: ORDERED, CLASSICAL and DISORDERED.

Supported DLCEP classes and data delivery features				
Item number	**Supported DLCEP class(es) and data delivery features**	**A) Reference**	**B) Status**	**C) Support**
1 (**A**)	CLASSICAL, ORDERED and UNORDERED peer and multi-peer DLCs, with DISORDERED DLCs either unsupported, or supported for both peer and multi-peer	11.2 b	C1	
2 (**B**)	only ORDERED and UNORDERED peer and multi-peer DLCs and CLASSICAL peer DLCs, with DISORDERED peer DLCs optional	11.2 b	C1	
3 (**C**)	only ORDERED and UNORDERED peer and multi-peer DLCs	11.2 b	C1	
4 (**D**)	only UNORDERED peer and multi-peer DLCs	11.2 b	C1	
C1 : exactly one of these alternatives must be chosen				

E.8.2.3 Support for subscribers-to-publisher DLSDUs

The implementation may provide the ability for subscribers on a multi-peer DLC to transmit DLSDUs to their publisher, providing a capability roughly equivalent to the inverse of a connectionless multi-cast.

Support for subscribers-to-publisher DLSDUs				
Item number	**Support for subscribers-to-publisher DLSDUs**	**A) Reference**	**B) Status**	**C) Support**
1	UNORDERED multi-peer DLCs	11.2 c	O	

E.8.2.4 Ratio of connection-mode maximum DLSDU size to maximum DLPDU-data size

The maximum-size DLSDU which can be transferred by an unordered DLC is the same as that which can be conveyed at the same priority by the connectionless data transfer service. ORDERED, CLASSICAL and DISORDERED DLCs provide the capability to transfer DLSDUs that are up to 16 times as large, by segmenting the DLSDU into 16 or fewer segments which will be transferred by separate DLPDUs. Thus the maximum ratio of DLSDU-size to DLPDU-data-size is 1 for UNORDERED DLCs and connectionless data transfer, and is between 1 and 16, inclusive for ORDERED, CLASSICAL and DISORDERED DLCs.

Supported maximum DLSDU : DLPDU-data ratios				
Item number	**maximum DLSDU-size/DLPDU-data size ratios**	**Reference**	**Allowed Range**	**Support**
1	CLASSICAL multi-peer DLCs	11.2 d	0:16 C1	
2	CLASSICAL peer DLCs	11.2 d	0:16 C1	
3	ORDERED peer and multi-peer DLCs	11.2 d	0:16	
4	UNORDERED peer and multi-peer DLCs	11.2 d	1	
C1 : **if** the supported value is non-zero **then** it must be equal to the value claimed in /3				

E.8.2.5 DL-timeliness

The implementation may provide the ability to record the DL-time at which a DLSDU is written to a buffer; to compute the timeliness of that buffer writing, or of subsequent buffer reading, or both, according to pre-established criteria; to convey that timeliness across a DLC; and to report both sending and receiving timeliness to a receiving DLS-user when delivering the DLSDU.

DL-timeliness support				
Item number	**Supported types of timeliness**	**A) Reference**	**B) Status**	**C) Support**
1	NONE	11.2 e	M	
2	TRANSPARENT	11.2 e	C1	
3	RESIDENCE	11.2 e	C2	
4	UPDATE	11.2 e	C2	
5	SYNCHRONIZED	11.2 e	C2	
C1 : **if** E.8.2.2/4.C **then** X **else** O C2 : **if** /2.C **then** O **else** X				

E.8.2.6 DL-time-stamped DLSDUs

The implementation may provide the ability to record the DL-time at which a DLSDU is written to a buffer by a DL-PUT request primitive and to convey that time across a DLC for subsequent use by a receiving DLE or for reporting to a receiving DLS-user when delivering the DLSDU.

DL-time-stamped DLSDU support				
Item number	**DL-time-stamped DLSDU support**	**A) Reference**	**B) Status**	**C) Sup-port**
1	NO	11.2 f	M	
2	YES	11.2 f	C1	
C1 : **if** E.8.2.5/2.C **then** O **else** X				

E.8.2.7 Connectionless data delivery features

The implementation must support unitdata transfer with local confirmation. It may also support unitdata transfer with remote confirmation, and unitdata exchange.

Supported connectionless data delivery features				
Item number	**Supported connectionless data delivery features**	**A) Reference**	**B) Status**	**C) Support**
1 (**X**)	unitdata exchange, and unitdata transfer with both local and remote confirmation	11.2 g	C1	
2 (**Y**)	only unitdata transfer with both local and remote confirmation	11.2 g	C1	
3 (**Z**)	only unitdata transfer with local confirmation	11.2 g	C1	
C1 : exactly one of these alternatives must be chosen				

E.8.3 DL-time and scheduling services

E.8.3.1 DL-time synchronism

The implementation may provide time synchronism with other devices on the extended Fieldbus.

Time synchronism support						
					Minimum required value for V(TDP)	
Item number	**Measure of provided time synchronism**	**A) Reference**	**B) Status**	**C) Support**	**D) Range**	**E) Support**
1	none	11.3 a	C1		none	
2	1 s	11.3 a	C1		Š 55 s C2	
3	100 ms	11.3 a	C1		Š 25 s C3	
4	10 ms	11.3 a	C1		Š 10 s C4	
5	1 ms	11.3 a	C1		Š 5 s C5	
6	100 ms	11.3 a	C1		Š 500 ms C6	
7	10 ms	11.3 a	C1		Š 50 ms C7	
8	1 ms	11.3 a	C1		Š 5 ms C8	

C1 : exactly one of these alternatives must be chosen

C2 : **if** /2.C **then** M **else** X

C3 : **if** /3.C **then** M **else** X

C4 : **if** /4.C **then** M **else** X

C5 : **if** /5.C **then** M **else** X

C6 : **if** /6.C **then** M **else** X

C7 : **if** /7.C **then** M **else** X

C8 : **if** /8.C **then** M **else** X

E.8.3.2 Time-based scheduling services

The implementation may support local time-based scheduling services.

Time-based scheduling service support				
Item number	**Support for PERIODIC schedules and ONE-TIME schedules with a specified starting time**	**A) Reference**	**B) Status**	**C) Support**
1	YES	11.3 b	C1	
2	NO	11.3 b	C2	

C1 :**if** time_synchronism_supported **then** C2 **else** X time_synchronism_supported = E.8.3.1/2 **or** /3 **or** /4 **or** /5 **or** /6 **or** /7 **or** /8

C2 : exactly one of these alternatives must be chosen

E.8.3.3 LAS time-based scheduling activities

The Fieldbus Data Link layer protocol requires at least one device on each local link to be functioning as a Link Master or a Bridge. Both Link Master and Bridge class devices are capable of initializing the token on the local link, and of serving as the local Link Active Scheduler (LAS). When functioning as an LAS, these devices provide centralized schedule-execution services, and may provide centralized schedule distribution and schedule construction services, for the entire link.

Bridge-class and Link Master-class devices can be full-capability devices, capable of creating a schedule for the local link, resources permitting, from any combination of scheduling requests received from the various devices on the link. Such devices instead may offer lesser capability, permitting construction of reduced-functionality networks. This reduced-functionality may be adequate for many devices which will become the LAS only if there are no bridges or higher-capability devices active on the link, or when the intended usage is only on preconfigured local links.

Five levels of scheduling support for Link Master-class and Bridge-class devices have been defined:

LAS time-based scheduling activity support				
Item number	**LAS time-based scheduling activity support**	**A) Reference**	**B) Status**	**C) Support**
1	DYNAMIC — can construct, distribute, receive and execute time-based schedules for the local link	11.3 c	C1	
2	SHARABLE — can distribute, receive and execute time-based schedules for the local link, but cannot construct such schedules	11.3 c	C1	
3	UPDATABLE — can receive and execute time-based schedules for the local link, but cannot distribute such schedules	11.3 c	C1	
4	STATIC — can execute time-based schedules for the local link, but cannot receive such schedules	11.3 c	C1	
5	NONE — no time-based scheduling for link	11.3 c	C2	
C1 **:if** time_synchronism_supported **then** C2 **else** X	time_synchronism_supported = E.8.3.1/2 **or** /3 **or** /4 **or** /5 **or** /6 **or** /7 **or** /8			
C2 : exactly one of these alternatives must be chosen				

E.9 Major low-level capabilities

This clause identifies many of the major conformance requirements of the implementation. The conformance requirements specified in this clause are those related to the lower-level medium-access functions of the DL-protocol. E.10 specifies the conformance requirements for the higher-level functions of the protocol — those which are independent of the underlying medium-access functions.

E.9.1 DLE classes implemented

The Fieldbus Data Link layer protocol defines three major lower-level classes of DLE operation, each a superset of the prior class. The support requirements for these classes differ.

DLE Classes Implemented				
Item number	**DLE Class**	**Reference**	**Status**	**Support**
1	Basic	5.6.1	M	
2	Link Master	5.6.2	O	
3	Bridge	5.6.3	C1	
C1 : **if** /2 **then** O **else** X				

E.9.2 Basic addressing requirements

DL-addresses have multiple lengths, and potentially have two synonyms. Where such synonyms exist, the choice of the appropriate synonym for use in a specific DLPDU is dependent both on the required QoS and on the availability of similar synonyms for all of the other addresses in the DLPDU.

Basic addressing requirements				
Item number	**Addressing requirements**	**Reference**	**Status**	**Support**
1	Are all LONG DL-addresses four octets in length?	6.2.2.1, A.1	M	
2	Are all SHORT DL-addresses two octets in length?	6.2.2.2, A.1	M	
3	Are all DL-addresses interpreted as specified in Tables A.1, A.2 and A.4 of Annex A?	A.2	M	
4	Are all predefined DL-addresses for the implemented classes claimed under E.9.1 recognized as specified in Tables A.5, A.6 and A.7 of Annex A?	A.3	M	
5	Are all LONG DL-addresses with a link-designator subfield (HL) whose value is not equal zero and is not equal V(TL) considered non-local?	A.2	M	
6	Are all LONG DL-addresses with a link-designator subfield (HL) whose value is zero or is V(TL), and with a node-designator subfield whose value is not equal zero and is not equal V(TN), considered link-local?	A.2	M	
7	Are all LONG DL-addresses with a link-designator subfield (HL) whose value is zero or is V(TL), and with a node-designator subfield whose value is equal zero or is equal V(TN), considered node-local?	A.2	M	
8	Are all SHORT DL-addresses with a node-designator subfield (N) whose value is not equal zero and is not equal V(TN), considered link-local?	A.2	M	
9	Are all SHORT DL-addresses with a node-designator subfield (N) whose value is equal zero or is equal V(TN), considered node-local?	A.2	M	
10	During address recognition, are all node-local DL-addresses with the same sub-node selector (S) considered equivalent?	A.2	M	
11	During address recognition, are all link-local DL-addresses with the same sub-link selector (N.S) considered equivalent?	A.2	M	
12	During DLPDU formation, when the DLS-user's DLPDU-authentication QoS was SOURCE or MAXIMAL, then is a source DL-address included within each resultant DLPDU, when possible?	5.2.4	M	
13	During DLPDU formation, when the DLS-user's DLPDU-authentication QoS was MAXIMAL, then are all DL-addresses within each resultant DLPDU LONG?	5.2.4	M	
14	During DLPDU formation, when at least one of the DLPDU's addresses is non-local, then are all DL-addresses within that DLPDU LONG?	6.2.1.1	M	
15	During DLPDU formation, when a DL-address is both link-local and LONG, then is the link component of the DL-address equal to the local link's link_id, and not zero unless that link_id is itself zero?	A.2.1	M	
16	During DLPDU formation, when a DL-address is node-local, then is the node component of the DL-address equal to the node's node_id, and not zero?	6.2.2.1, A.2.2	M	
17	During DLPDU formation, when the DLS-user's DLPDU-authentication QoS was ORDINARY or SOURCE, and all DL-addresses within the DLPDU are link-local, then are those DL-addresses all SHORT?	5.2.4	M	

E.9.3 DLPDU support

This sub-clause enumerates the types of DLPDU which the implementation can receive and recognize, and, separately, which the implementation can transmit, as a function of the highest conformance class claimed under E.9.1; and separately as a function of the momentary role within the local data link assumed by the implementation.

NOTE — It is assumed that an implementation can receive (in a rudimentary sense) all DLPDUs, whether well-formed or not, but that well-formed DLPDUs which are not recognized are ignored.

E.9.3.1 Static DLPDU support

The Fieldbus Data Link layer protocol defines three major classes of operation, each a superset of the prior class. This table specifies the static (that is, independent of momentary role) conformance requirements as a function of the claimed capabilities of the implementation. Only the entries for the highest claimed class (right-most in the table) need be completed.

Class dependency of Supported DLPDUs															
			Basic Class				Link Master Class				Bridge Class				
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending		C) Receiving and Recognizing		D) Forming and Sending		E) Receiving and Recognizing		F) Forming and Sending	
Item number	**DLPDU type**	**Reference**	**Stat -us**	**Sup- port**	**Stat -us**	**Sup- port**	**Stat -us**	**Sup- port**	**Stat -us**	**Sup- port**	**Stat -us**	**Sup- port**	**Stat -us**	**Sup- port**
1	EC	7.1	M1		M1		M1		M1		M1		M1	
2	DC	7.2	M1		M1		M1		M1		M1		M1	
3	RC	7.3	M1		C2		M1		C2		M1		M1	
4	CA	7.4	M1		C3		M1		C3		M		M	
5	CD	7.5	M		M		M		M		M		M	
6	ED	7.6	M1		C4		M1		C4		M		M	
7	DT	7.7	M		M		M		M		M		M	
8	SR	7.8	O		M		M		M		M		M	
9	CT	7.9	X		C5		M		C5		M		C5	
10	TD	7.10	C5		X		C5		M		C5		M	
11	RQ	7.11	M		C5		M		C5		M		C5	
12	RR	7.12	C5		M		C5		M		C5		M	
13	PN	7.13	M		X		M		M		M		M	
14	PR	7.14	M		M		M		M		M		M	
15	PT	7.15	M		X		M		M		M		M	
16	ES	7.16	M1		X		M1		M		M1		M	
17	RT	7.17	X		M		M		M		M		M	
18	RI	7.18	X		O		M		O		M		O	
19	CL	7.19	X		X		M		M		M		M	
20	TL	7.20	X		X		M		M		M		M	
21	WK	7.22	O		X		O		M		X		M	
22	IDLE	7.23	X		O		X		O		X		O	
23	spare	7.24	X		X		X		X		X		X	
24	don't use	7.25	X		X		X		X		X		X	
M1 : only a minimal subset need be implemented														
C2 : **if** reliable_CO **then** M **else** X								reliable_CO = E.8.2.2/1 **or** /2						
C3 : **if** acknowledged_CL **then** M **elseif** E.8.2.3/1 **then** O **else** X								acknowledged_CL = E.8.2.7/1 **or** /2						
C4 : **if** E.8.2.7/1 **then** M **else** O														
C5 : **if** time_synchronism_supported **then** M **else** O								time_synchronism_supported = E.8.3.1/2 **or** /3 **or** /4 **or** /5 **or** /6 **or** /7 **or** /8						

E.9.3.2 Dynamic DLPDU support

The Fieldbus Data Link layer protocol defines four major roles for operating devices. This table specifies the dynamic (that is, dependent on momentary role) conformance requirements of the implementation as a function of the then-appropriate role. For an implementation claiming only Basic Class conformance, only the left-most four columns (A–D) need to be completed. For all other implementations, all columns need to be completed.

Role dependency of Supported DLPDUs																		
			when DLE’s operational Node address is unknown and DLE has assumed a temporary Node address				when DLE’s operational Node address is known and DLE is functioning as a Basic DLE				when DLE’s operational Node address is known and DLE is functioning as a Link Master or Bridge DLE and DLE is not acting as the LAS				when DLE’s operational Node address is known and DLE is functioning as a Link Master or Bridge DLE and DLE is acting as the LAS			
	Column letter →		A) Receiving and Recog-nizing		B) Forming and Send-ing		C) Receiving and Recog-nizing		D) Forming and Send-ing		E) Receiving and Recog-nizing		F) Forming and Send-ing		G) Receiving and Recog-nizing		H) Forming and Sending	
Item num-ber	**DLPDU type**	**Reference**	**Stat-us**	**Sup-port**	**Stat-us**	**Sup-port**	**Stat-us**	**Sup-port**	**Stat-us**	**Sup-port**	**Stat-us**	**Sup-port**	**Stat-us**	**Sup-port**	**Stat-us**	**Sup-port**	**Stat-us**	**Sup-port**
1	EC	7.1	O		X		M1		M1		M1		M1		M1		M1	
2	DC	7.2	O		X		M1		M1		M1		M1		M1		M1	
3	RC	7.3	O		X		M1		C2		M1		C2		M1		M1	
4	CA	7.4	M1		X		M1		C3		M1		C3		M		M	
5	CD	7.5	M		X		M		M		M		M		M		M	
6	ED	7.6	M1		X		M1		C4		M1		C4		M		M	
7	DT	7.7	M		X		M		M		M		M		M		M	
8	SR	7.8	O		X		M		M		M		M		M		M	
9	CT	7.9	X		X		X		C5		M		C5		M		C5	
10	TD	7.10	C5		X		C5		X		C5		M		C5		M	
11	RQ	7.11	X		X		M		C5		M		C5		M		C5	
12	RR	7.12	X		X		C5		M		C5		M		C5		M	
13	PN	7.13	M		X		M		X		M		X		X		M	
14	PR	7.14	M		M		M		M		M		M		M		X	
15	PT	7.15	M		X		M		X		M		X		X		M	
16	ES	7.16	X		X		M1		X		M1		X		X		M	
17	RT	7.17	X		X		X		M		X		M		M		X	
18	RI	7.18	X		X		X		O		X		O		M		X	
19	CL	7.19	X		X		X		X		M		M		M		X	
20	TL	7.20	X		X		X		X		M		X		X		M	
21	WK	7.22	X		X		M		X		M		X		X		M	
22	IDLE	7.23	X		X		X		O		X		O		X		O	
23	spare	7.24	X		X		X		X		X		X		X		X	
	don’t use	7.25	X		X		X		X		X		X		X		X	

M1 : only a minimal subset need be implemented

C2 : **if** reliable_CO **then** M **else** X — reliable_CO = E.8.2.2/1 **or** /2

C3 : **if** acknowledged_CL **then** M
elseif E.8.2.3/1 **then** O **else** X — acknowledged_CL = E.8.2.7/1 **or** /2

C4 : **if** E.8.2.7/1 **then** M **else** O

C5 : **if** time_synchronism_supported **then** M **else** O — time_synchronism_supported = E.8.3.1/2 **or** /3 **or** /4 **or** /5 **or** /6 **or** /7 **or** /8

E.9.4 DLPDU-class specific fields and subfields

This sub-clause enumerates the ranges and states the static conformance requirements for each DLPDU-class-specific field and subfield of each DLPDU. Conformance is only required on those DLPDU classes which were claimed under E.9.3.1 and E.9.3.2, and then only for the direction(s) — receiving and recognizing, or forming and sending, or both — which were claimed. Thus any conditions or options specified in the relevant entry or entries in E.9.3.1 and E.9.3.2 apply uniformly to all of the specific requirements of the DLPDU class, as detailed in the following subclauses, for the corresponding direction of use. Requirements in the following DLPDU-specific subclauses for directions of use not claimed under either E.9.3.1 or E.9.3.2 can be ignored.

E.9.4.1 Specific fields and subfields of the EC DLPDU

Supported Parameters						
		Column letter →	**A) Receiving and Recognizing**		**B) Forming and Sending**	
Item number	**Parameter**	**Reference**	**Status or Range**	**Support**	**Status or Range**	**Support**
1	address size designator	6.2.1.1,				
1.1	LONG	7.1.1.1	C1		C1	
1.2	SHORT		M		M	
2	final token use designator	6.2.1.4,				
2.1	NOT-FINAL	7.1.1.1	M		M	
2.2	FINAL		M		M	
3	DL address formats:	6.2.2,				
3.1	format 1L — LONG destination and two sources	7.1.1.1, 7.1.1.2	M		M	
3.2	format 1S — SHORT destination and two sources		M		M	
3.3	format 2L — two LONG sources only		M		M	
3.4	format 2S — two SHORT sources only		M		M	
4	EC-parameter:	7.1.1.3,				
4.0	version 1	8.1	M		M	
5	permitted octets of user data in DLPDU	6.2.4, 7.1.1.4	0 : 128 octets M		0 : 128 octets M	
C1 : **if** E.8.1/1 **then** M **else** X						

E.9.4.2 Specific fields and subfields of the DC DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status or Range	Support	Status or Range	Support
1	address size designator	6.2.1.1,				
1.1	LONG	7.2.1.1	C1		C1	
1.2	SHORT		M		M	
2	final token use designator	6.2.1.4,				
2.1	NOT-FINAL	7.2.1.1	M		M	
2.2	FINAL		M		M	
3	DL address formats:	6.2.2,				
3.1	format 1L — LONG destination and source	7.2.1.1, 7.2.1.2	M		M	
3.2	format 1S — SHORT destination and source		M		M	
3.3	format 2L — LONG source only		M		M	
3.4	format 2S — SHORT source only		M		M	
4	DC-parameter:	7.2.1.3,				
4.0	version 1	8.2	M		M	
5	permitted octets of user data in DLPDU	6.2.4, 7.2.1.4	0 : 128 octets M		0 : 128 octets M	
C1 : **if** E.8.1/1 **then** M **else** X						

E.9.4.3 Specific fields and subfields of the RC DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status or Range**	**Support**	**Status or Range**	**Support**
1	address size designator	6.2.1.1,				
1.1	LONG	7.3.1.1	C1		C1	
1.2	SHORT		M		M	
2	final token use designator	6.2.1.4,				
2.1	NOT-FINAL	7.3.1.1	M		M	
2.2	FINAL		M		M	
3	priority designator	6.2.1.2,				
3.1	URGENT	7.3.1.1	C2		C2	
3.2	NORMAL		C2		C2	
3.3	TIME-AVAILABLE		M		M	
4	DL address formats:	6.2.2,				
4.1	format 1L — LONG destination and source	7.3.1.1, 7.3.1.2	M		M	
4.2	format 1S — SHORT destination and source		M		M	
4.3	format 2L — LONG source only		M		M	
4.4	format 2S — SHORT source only		M		M	
5	RC-parameter:	7.3.1.3,				
5.0	version 1	8.3	M		M	
Item number	**Parameter**	**Reference**	**Range (in octets)**	**Supported Range**	**Range (in octets)**	**Supported Range**
6	permitted octets of user data in DLPDU	6.2.4,				
6.1	URGENT priority	7.5.1.4	0 : 64 C2		0 : 64 C2	
6.2	NORMAL priority		0 : 128 C2		0 : 128 C2	
6.3	TIME-AVAILABLE priority		0 : 256 M		0 : 256 M	
C1 : **if** E.8.1/1 **then** M **else** X						
C2 : **if** E.8.2.1/2 **then** M **else** x						

E.9.4.4 Specific fields and subfields of the CA DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status**	**Support**	**Status**	**Support**
1	address size designator	6.2.1.1,				
1.1	LONG	7.4.1.1	C1		C1	
1.2	SHORT		M		M	
2	final token use designator	6.2.1.4,				
2.1	NOT-FINAL	7.4.1.1	M		M	
2.2	FINAL		M		M	
3	priority designator	6.2.1.2,				
3.1	URGENT	7.4.1.1	M2		C3	
3.2	NORMAL		M2		C3	
3.3	TIME-AVAILABLE		M		M	
4	DL address formats	6.2.2,				
4.1	format 1L — LONG destination and source	7.4.1.1, 7.4.1.2	M		M	
4.2	format 1S — SHORT destination and source		M		M	
4.3	format 2L — LONG destination, omitted source		M		M	
4.4	format 2S — SHORT destination, omitted source		M		M	
5	SD-parameter:	7.4.1.3,				
5.0	version 1 supported	8.4	M		M	
Item number	**Parameter**	**Reference**	**Range (in octets)**	**Supported Range**	**Range (in octets)**	**Supported Range**
6	permitted octets of user data in DLPDU	6.2.4,				
6.1	URGENT priority	7.5.1.4	0 : 64 M2		0 : 64 C3	
6.2	NORMAL priority		0 : 128 M2		0 : 128 C3	
6.3	TIME-AVAILABLE priority		0 : 256 M		0 : 256 M	

C1 : **if** E.8.1/1 **then** M **else** X

M2 : **if** E.8.2.1/2 **then** M **else** only a minimal subset need be implemented

C3 : **if** E.8.2.1/2 **then** M **else** X

E.9.4.5 Specific fields and subfields of the CD DLPDU

Supported Parameters						
Item number	**Column letter →**		**A) Receiving and Recognizing**		**B. Forming and Sending**	
	Parameter	**Reference**	**Status**	**Support**	**Status**	**Support**
1	address size designator	6.2.1.1,				
1.1	LONG	7.5.1.1	C1		C1	
1.2	SHORT		M		M	
2	final token use designator	6.2.1.4,				
2.1	NOT-FINAL	7.5.1.1	M		M	
2.2	FINAL		M		M	
3	priority designator	6.2.1.2,				
3.1	URGENT	7.5.1.1	M2		C3	
3.2	NORMAL		M2		C3	
3.3	TIME-AVAILABLE		M		M	
4	DL address formats	6.2.2,				
4.1	format 1L — LONG destination and source	7.5.1.1,	M		M	
		7.5.1.2				
4.2	format 1S — SHORT destination and source		M		M	
4.3	format 2L — LONG destination, omitted source		M		M	
4.4	format 2S — SHORT destination, omitted source		M		M	
5	SD-parameter:	7.5.1.3,				
5.0	version 1 supported	8.4	M		M	
6	user data in DLPDU	7.5.1.4	X		X	

C1 : **if** E.8.1/1 **then** M **else** X

M2 : **if** E.8.2.1/2 **then** M **else** only a minimal subset need be implemented

C3 : **if** E.8.2.1/2 **then** M **else** X

E.9.4.6 Specific fields and subfields of the ED DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status**	**Support**	**Status**	**Support**
1	address size designator	6.2.1.1,				
1.1	LONG	7.6.1.1	C1		C1	
1.2	SHORT		M		M	
2	final token use designator	6.2.1.4,				
2.1	NOT-FINAL	7.6.1.1	M		M	
2.2	FINAL		M		M	
3	priority designator	6.2.1.2,				
3.1	URGENT	7.6.1.1	M2		C3	
3.2	NORMAL		M2		C3	
3.3	TIME-AVAILABLE		M		M	
4	DL address formats	6.2.2,				
4.1	format 1L — LONG destination and source	7.6.1.1, 7.6.1.2	M		M	
4.2	format 1S — SHORT destination and source		M		M	
4.3	format 2L — LONG destination, omitted source		M		M	
4.4	format 2S — SHORT destination, omitted source		M		M	
5	SD-parameter:	7.6.1.3,				
5.0	version 1 supported	8.4	M		M	
Item number	**Parameter**	**Reference**	**Range (in octets)**	**Supported Range**	**Range (in octets)**	**Supported Range**
6	permitted octets of user data in DLPDU	6.2.4,				
6.1	URGENT priority	7.5.1.4	0 : 64 M2		0 : 64 C3	
6.2	NORMAL priority		0:128 M2		0:128 C3	
6.3	TIME-AVAILABLE priority		0 : 256 M		0 : 256 M	

C1 : **if** E.8.1/1 **then** M **else** X

M2 : **if** E.8.2.1/2 **then** M **else** only a minimal subset need be implemented

C3 : **if** E.8.2.1/2 **then** M **else** X

E.9.4.7 Specific fields and subfields of the DT DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status	Support	Status	Support
1	address size designator	6.2.1.1,				
1.1	LONG	7.7.1.1	C1		C1	
1.2	SHORT		M		M	
2	final token use designator	6.2.1.4,				
2.1	NOT-FINAL	7.7.1.1	M		M	
2.2	FINAL		M		M	
3	priority designator	6.2.1.2,				
3.1	URGENT	7.7.1.1	M2		C3	
3.2	NORMAL		M2		C3	
3.3	TIME-AVAILABLE		M		M	
4	DL address formats	6.2.2,				
4.1	format 1L — LONG destination and source	7.7.1.1, 7.7.1.2	M		M	
4.4	format 1S — SHORT destination and source		M		M	
4.2	format 2L — LONG destination, omitted source		M		M	
4.5	format 2S — SHORT DESTINATION, omitted source		M		M	
4.3	format 3L — long SOURCE only		M		M	
4.6	format 3S — short SOURCE only		M		M	
4.7	format 4 — omitted DESTINATION and omitted source		M		M	
4.8	format 5 — omitted SOURCE only		M		M	
5	SD-parameter:	7.5.1.3,				
5.0	version 1 SUPPORTED	8.3	M		M	
Item number	**Parameter**	**Reference**	**Range (in octets)**	**Supported Range**	**Range (in octets)**	**Supported Range**
6	permitted octets of user data in DLPDU	6.2.4,				
6.1	URGENT priority	7.5.1.4	0 :64 M2		0 : 64 C3	
6.2	NORMAL priority		0 :128 M2		0 : 128 C3	
6.3	TIME-AVAILABLE priority		0 : 256 M		0 : 256 M	

C1 : **if** E.8.1/1 **then** M **else** X

M2 : **if** E.8.2.1/2 **then** M **else** only a minimal subset need be implemented

C3 : **if** E.8.2.1/2 **then** M **else** X

E.9.4.8 Specific fields and subfields of the SR DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status and Range	Support	Status and Range	Support
1	DL address format: omitted destination, NODE source	6.2.2.3, 7.8.1.2	M (16: 255)		M (16: 255)	
2	SR-parameter	7.8.1.3	M		M	

E.9.4.9 Specific fields and subfields of the CT DLPDU

The CT DLPDU has no variable fields or subfields.

E.9.4.10 Specific fields and subfields of the TD DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status and Range	Support	Status and Range	Support
1	DL address format: NODE source, omitted destination	6.2.2.3, 7.10.1.2	M (16: 255)		M (16: 255)	
2	TD-parameter	7.10.1.3	M		M	

E.9.4.11 Specific fields and subfields of the RQ DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status and Range	Support	Status and Range	Support
1	DL address format: NODE.0 destination and source	6.2.2.3, 7.11.1.2	M (4,16: 255).0		M (4,16: 255).0	
2	RQ-parameter	7.11.1.3	M		M	

E.9.4.12 Specific fields and subfields of the RR DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status and Range	Support	Status and Range	Support
1	DL address format: NODE.0 destination and source	6.2.2.3, 7.12.1.2	M (4,16: 255).0		M (4,16: 255).0	
2	RR-parameter	7.12.1.3, 7.12.2	M		M	

E.9.4.13 Specific fields and subfields of the PN DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status	Support	Status	Support
1	DL address format: NODE destination, omitted source	6.2.1.1, 7.13.1.2	M (16: 255)		M (16: 255)	
2	PN-parameters	7.13.1.3, 7.13.2				

E.9.4.14 Specific fields and subfields of the PR DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Range (in octets)	Supported Range	Range (in octets)	Supported Range
1	permitted octets of SPDU in DLPDU	6.2.4, 7.14.1.4	0 : 64 M		0 : 64 M	

E.9.4.15 Specific fields and subfields of the PT DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status	Support	Status	Support
1	priority designator	6.2.1.2,				
1.1	URGENT	7.15.1.1	M		M	
1.2	NORMAL		M		M	
1.3	TIME-AVAILABLE		M		M	
2	DL address format: NODE destination, omitted source	6.2.1.1, 7.15.1.2	M (16: 255)		M (16: 255)	
3	DD-parameters	7.15.1.3, 7.15.2	M		M	

E.9.4.16 Specific fields and subfields of the ES DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status and Range	Support	Status and Range	Support
1	address size designator	6.2.1.1,				
1.1	LONG	7.17.1.1	C1		C1	
1.2	SHORT		M		M	
2	restart/continue designator	6.2.1.3, 7.17.1.1				
2.1	RESTART		M		M	
2.2	CONTINUE		M		M	
3	DL address formats:	6.2.2,				
3.1	format 1L — LONG destination, omitted source	7.17.1.1, 7.17.1.2	M		M	
3.2	format 1S — SHORT destination, omitted source		M		M	
4	DD-parameters	7.17.1.3	M		M	
C1 : **if** E.8.1/1 **then** M **else** X						

E.9.4.17 Specific fields and subfields of the RT DLPDUs

The RT DLPDU has no variable fields or subfields.

E.9.4.18 Specific fields and subfields of the RI DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status**	**Support**	**Status**	**Support**
1	DD-parameters	7.18.1.3	**M**		**M**	

E.9.4.19 Specific fields and subfields of the CL DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status and Range**	**Support**	**Status and Range**	**Support**
1	DL address format: omitted destination, NODE source	6.2.2.3, 7.19.1.2	M (16: 255)		M (16: 255)	

E.9.4.20 Specific fields and subfields of the TL DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status and Range**	**Support**	**Status and Range**	**Support**
1	DL address format: NODE destination, omitted source	6.2.2.3, 7.20.1.2	M (16: 255)		M (16: 255)	
			Range (in octets)	**Supported Range**	**Range (in octets)**	**Supported Range**
2	permitted octets of SPDU in DLPDU	6.2.4, 7.20.1.4	0 : 64 M		0 : 64 M	

E.9.4.21 Specific fields and subfields of the WK DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status and Range	Support	Status and Range	Support
1	DL address format: NODE destination, omitted source	6.2.2.3, 7.21.1.2	M (16: 255)		M (16: 255)	

E.9.4.22 Specific fields and subfields of the Idle DLPDU

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Range (in octets)	Supported Range	Range (in octets)	Supported Range
1	permitted octets of user data in DLPDU	6.2.4, 7.22.1.4	0 : 64 O		0 : 64 O	

E.10 Major high-level capabilities

This clause identifies many of the major conformance requirements of the implementation. The conformance requirements specified in this clause are those related to the higher-level medium-access-independent functions of the DL-protocol. E.9 specifies the conformance requirements for the lower-level functions of the protocol — those which are related to the underlying medium-access functions.

E.10.1 DLCEP classes implemented

The Fieldbus Data Link layer protocol defines four major classes of DLCEP operation, each a superset of the prior class. The support requirements for these classes differ.

DLCEP Classes Implemented					
Item number	DLCEP class	DLC profiles	Reference	Status	Support
1	UNORDERED peer, publisher and subscriber DLCs	A, B, C, D	9.2, 11.2	M	
2	ORDERED peer, publisher and subscriber DLCs	A, B, C	9.2, 11.2	O	
3	CLASSICAL peer DLCs	A, B	9.2, 11.2	C1	
4	CLASSICAL publisher and subscriber DLCs	A	9.2, 11.2	C2	
C1 : support of this class requires support of ORDERED peer, publisher and subscriber DLCEPs C2 : support of this class requires support of CLASSICAL peer DLCEPs					

E.10.2 Parameters of the EC, DC, RC, CA, CD, ED and DT DLPDUs

This sub-clause enumerates the ranges and states the static conformance requirements for each variable parameter of each EC, DC, RC, CA, CD, ED and DT DLPDU. Conformance is only required on those DLPDUs which were claimed under E.9.3.1 and E.9.3.2, and then only for the direction(s) — receiving and recognizing, or forming and sending, or both — which were claimed. Thus any conditions or options specified in the relevant entry or entries in E.9.3.1 and E.9.3.2 apply uniformly to all of the DLPDU's specific requirements, as detailed in the following subclauses, for the corresponding direction of use. Requirements in the following DLPDU-specific subclauses for directions of use not claimed under either E.9.3.1 or E.9.3.2 can be ignored.

E.10.2.1 Communications paradigms implemented

The Fieldbus Data Link layer protocol specifies a number of communications paradigms. The support requirements for many of these paradigms differ.

Communications Paradigms Implemented				
Item number	**Communications Paradigm**	**Reference**	**Status**	**Support**
1	Unacknowledged Connectionless	5.5.2, 9.3	M	
2	Acknowledged Connectionless	5.5.2, 9.3	C1	
3	Connectionless Exchange	5.5.2, 9.3	C2	
4	Unordered Connection as Peer	5.5.3, 9.2, 11.2	M	
5	Ordered Connection as Peer	5.5.3, 9.2, 11.2	C3	
6	Classical Connection as Peer	5.5.3, 9.2, 11.2	C4	
7	Disordered Connection as Peer	5.5.3, 9.2, 11.2	C6	
8	Unordered Connection as Publisher	5.5.3, 9.2, 11.2	M	
9	Ordered Connection as Publisher	5.5.3, 9.2, 11.2	C3	
10	Classical Connection as Publisher	5.5.3, 9.2, 11.2	C5	
11	Disordered Connection as Publisher	5.5.3, 9.2, 11.2	C7	
12	Unordered Connection as Subscriber	5.5.3, 9.2, 11.2	M	
13	Ordered Connection as Subscriber	5.5.3, 9.2, 11.2	C3	
14	Classical Connection as Subscriber	5.5.3, 9.2, 11.2	C5	
15	Disordered Connection as Subscriber	5.5.3, 9.2, 11.2	C7	

C1 : **if** E.8.2.7/3 **then** X **else** M.

C2 : **if** E.8.2.7/1 **then** M **else** X.

C3 : **if** DLC_conformance_class = {A, B, or C} **then** M **else** O.

C4 : **if** DLC_conformance_class = {A or B} **then** M **else** O.

C5 : **if** DLC_conformance_class = A **then** M **else** O.

C6 : **if** C4 **then** O **else** X.

C7 : **if** C5 **and** C6 **then** M **else** X.

E.10.2.2 Support for implied QoS for data link connections (DLCs)

Some aspects of data link connections also impact on conformance requirements.

Support for implied QoS (Quality of Service) parameters for data link connections							
		Column letter →		**A) Receiving Data**		**B) Sending Data**	
Item number	**QoS Parameter**	**Reference**	**Allowed Range**	**Status**	**Support**	**Status**	**Support**
1	segmentation — what is the ratio of the maximum DLSDU size to the maximum DLPDU size?	11.2	1 : 16	C1		C1	
2	what is the maximum window size on CLASSICAL DLCs?	8.1 Table 27	1 : 15	C2		C2	

C1 : **if** DLC_conformance_class = {A, B, or C} **then** 1:16 M **else** 1 M;
the same value must be specified for both columns.

C2 : **if** DLC_conformance_class = {A or B} **then** M **else** O; the same value must be specified for both columns.

E.10.2.3 Parameters of the EC DLPDU — protocol version 1

E.10.2.3.1 DLC basic attributes

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status or Range**	**Support**	**Status or Range**	**Support**
1	DLC protocol version number	8.1 a4	1 M		1 M	
2	reply requested	8.1 a1	F, T M		F, T M	
3	publisher-DLCEP-address reuse-discriminator	8.1 a2	F, T M		F, T M	
4	DLL path diversity	8.1 a3				
4.1	ANY-PATH		M		M	
4.2	THIS-PATH		M		M	
5	DLCEP class	8.1 a5				
5.1	peer DLCEP		M		M	
5.2	publisher DLCEP		M		M	
5.3	subscriber DLCEP		M		M	
6	DLL priority	8.1 a6				
6.1	URGENT		M		O	
6.2	NORMAL		M		O	
6.3	TIME-AVAILABLE		M		M	
7	DL address size	8.1 a7				
7.1	LONG		M		M	
7.2	SHORT		M		O	
7.3	VERY-SHORT		M		O	
8	DLPDU authentication	8.1 a8				
8.1	MAXIMAL		M		M	
8.2	SOURCE		M		M	
8.3	ORDINARY		M		M	
9	maximum DL-CONNECT, DL-RESET and DL-SUBSCRIBER-QUERY confirm delay	8.1 b				
9.1	UNLIMITED (encoded as 0xFFFF)		65 535 M		65 535 M	
9.2	minimum delay required (in ms)		1 : 60 000		1 : 60 000	
9.3	maximum delay supported (in ms)		1 : 60 000 M		1 : 60 000 M	
10	maximum DL-DATA confirm delay	8.1 b				
10.1	UNLIMITED (encoded as 0xFFFF)		65 535 M		65 535 M	
10.2	minimum delay required (in ms)		1 : 60 000		1 : 60 000	
10.3	maximum delay supported (in ms)		1 : 60 000 M		1 : 60 000 M	

E.10.2.3.2 DLC attributes as sender (part 1)

Supported Parameters						
		Column letter →	**A) Receiving and Recognizing**		**B) Forming and Sending**	
Item number	**Parameter**	**Reference**	**Status or Range**	**Support**	**Status or Range**	**Support**
1	DLC data delivery features	8.1 c1				
1.1	UNORDERED — peer, publisher and subscriber		M		M	
1.2	ORDERED — peer, publisher and subscriber		M		C1	
1.3	CLASSICAL — peer		M		C2	
1.4	DISORDERED — peer		M		O	
1.5	CLASSICAL — publisher and subscriber		M		C3	
1.6	DISORDERED — publisher and subscriber		M		C4	
2	DLC residual activity required	8.1 c2				
2.1	FALSE (no background activity)		M		M	
2.2	TRUE (background activity)		M		M	
3	DLSDU storage model at DLCEP	8.1 c3				
3.1	QUEUE		M		M	
3.2	BUFFER		M		M	
4	DLC maximum window size	8.1 c4	0:15 M		0:15 M1	

C1 : **if** DLC_conformance_class = {A, B, or C} **then** M **else** O.

C2 : **if** DLC_conformance_class = {A or B} **then** M **else** O.

C3 : **if** DLC_conformance_class = A **then** M **else** O.

C4 : **if** C3 and disordered_peer **then** M **else** O.

disordered_peer = /1.4

M1 : must be 1:*N*, where *N* ³ the value claimed in E.10.2.2/2

E.10.2.3.3 DLC attributes as sender (part 2)

Supported Parameters (cont.)						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status or Range**	**Support**	**Status or Range**	**Support**
5	SD-parameter basic-DLC-parameters format	8.1 c5				
5.1	A – NULL format		M		C1	
5.2	B – SHORT format		M		C1	
5.3	C – LONG format		M		C1	
5.4	D – UNSEGMENTED LONG format		M		C1	
5.5	E – SUBSCRIBER format		M		C1	
5.6	F – PUBLISHER format		M		C1	
5.7	G – UNSEGMENTED PUBLISHER format		M		C1	
6	2-way data exchange permitted	8.1 c6				
6.1	FALSE		M		M	
6.2	TRUE		M		O	
7	timeliness included in DLPDU	8.1 c8				
7.1	FALSE		M		M	
7.2	TRUE		M		C5	
8	SD-parameter time-stamp format	8.1 c9				
8.1	J – NULL format		M		M	
8.2	K – two-octet format		M		C6	
8.3	L – three-octet format		M		C6	
8.4	M – six-octet format		M		C6	
9	DLC maximum DLSDU size	8.1 c10	0:4096 M		0:4096 M2	
C1 : **if** DLC_conformance_class = {A, B, or C} **then** M **else** O.						
C5 : **if** DLC_conformance_class = {A or B} **then** M **else** O.						
C6 : **if** DLC_conformance_class = A **then** M **else** O.						
M2 : must be 1:N, where N = 256 times the value claimed in E.10.2.2/1						

E.10.2.3.4 DLC attributes as receiver (part 1)

Supported Parameters						
	Column letter →		**A) Receiving and Recognizing**		**B) Forming and Sending**	
Item number	**Parameter**	**Reference**	**Status or Range**	**Support**	**Status or Range**	**Support**
1	DLC data delivery features	8.1 c1				
1.1	UNORDERED — peer, publisher and subscriber		M		M	
1.2	ORDERED — peer, publisher and subscriber		M		C1	
1.3	CLASSICAL — peer		M		C2	
1.4	DISORDERED — peer		M		O	
1.5	CLASSICAL — publisher and subscriber		M		C3	
1.6	DISORDERED — publisher and subscriber		M		C4	
2	DLC residual activity required	8.1 c2				
2.1	FALSE (no background activity)		M		M	
2.2	TRUE (background activity)		M		M	
3	DLSDU storage model at DLCEP	8.1 c3				
3.1	QUEUE		M		M	
3.2	BUFFER		M		M	
4	DLC maximum window size	8.1 c4	0:15 M		0:15 M1	

C1 : **if** DLC_conformance_class = {A, B, or C} **then** M **else** O.

C2 : **if** DLC_conformance_class = {A or B} **then** M **else** O.

C3 : **if** DLC_conformance_class = A **then** M **else** O.

C4 : **if** C3 and disordered_peer **then** M **else** O.

disordered_peer = /1.4

M1 : must be 1:N, where N ³ the value claimed in E.10.2.2/2

E.10.2.3.5 DLC attributes as receiver (part 2)

Supported Parameters (cont.)						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status or Range	Support	Status or Range	Support
5	SD-parameter basic-DLC-parameters format	8.1 c5				
5.1	A – NULL format		M		C1	
5.2	B – SHORT format		M		C1	
5.3	C – LONG format		M		C1	
5.4	D – UNSEGMENTED LONG format		M		C1	
5.5	E – SUBSCRIBER format		M		C1	
5.6	F – PUBLISHER format		M		C1	
5.7	G – UNSEGMENTED PUBLISHER format		M		C1	
6	2-way data exchange permitted	8.1 c6				
6.1	FALSE		M		M	
6.2	TRUE		M		O	
7	timeliness included in DLPDU	8.1 c8				
7.1	FALSE		M		M	
7.2	TRUE		M		C5	
8	SD-parameter time-stamp format	8.1 c9				
8.1	J – NULL format		M		M	
8.2	K – two-octet format		M		C6	
8.3	L – three-octet format		M		C6	
8.4	M – six-octet format		M		C6	
9	DLC maximum DLSDU size	8.1 c10	0:4096M		0:4096 M2	

C1 : **if** DLC_conformance_class = {A, B, or C} **then** M **else** O.
C5 : **if** DLC_conformance_class = {A or B} **then** M **else** O.
C6 : **if** DLC_conformance_class = A **then** M **else** O.
M2 : must be 1:*N*, where *N* = 256 times the value claimed in E.10.2.2/1

E.10.2.4 Parameters of the DC DLPDU — protocol version 1

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status or Range	Support	Status or Range	Support
1	DLC protocol version number	8.2 a3	1 M		1 M	
2	reply requested	8.2 a1	F, T M		F, T M	
3	reason for requested action	8.2 b	M		M	

E.10.2.5 Parameters of the RC DLPDU — protocol version 1

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status or Range	Support	Status or Range	Support
1	DLC protocol version number	8.3 a, 8.2.a3	1 M		1 M	
2	reply requested	8.3 a, 8.2.a1	F, T M		F, T M	
3	reason for requested action	8.3 b	M		M	
4	sequence number residue	8.3 c	M		M	

E.10.2.6 SD-parameters of the CA, CD, DT and ED DLPDUs — protocol version 1

E.10.2.6.1 Unitdata transaction-support parameters

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status	Support	Status	Support
1	SD-parameter	8.4.1				
	connectionless-mode parameters formats					
1.1	format P — NULL	8.4.1 a	M		M	
1.2	format R — INITIATOR	8.4.1 b	M		C1	
1.3	format U — RESPONDER	8.4.1 c	C1		M	
C1 **if** E.8.2.7/3 **then** X **else** M						

E.10.2.6.2 Basic DLC parameters (part 1)

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status**	**Support**	**Status**	**Support**
1	SD-parameter basic-DLC-parameters formats	8.4.2.1				
1.1	format A — NULL	8.4.2.1 A	M		M	
1.2	format B — SHORT	8.4.2.1 B	C1		C1	
1.2.1	retransmit	8.4.2.1 B1	F, T C1		F, T C1	
1.2.2	acknowledge	8.4.2.1 B2	F, T C1		F, T C1	
1.2.3	timeliness	8.4.2.1 B3	F, T C1		F, T C1	
1.2.4	modulus number of post-acknowledged or requested DLSDU	8.4.2.1 B4	C1		C1	
1.2.5	modulus number of associated DLSDU or of highest-numbered DLSDU sent	8.4.2.1 B5	C1		C1	
1.3	format C — LONG	8.4.2.1 C	C1		C1	
1.3.1	modulus number of post-acknowledged or requested DLSDU	8.4.2.1 C1	C1		C1	
1.3.2	(zero-origin) requested segment number	8.4.2.1 C2	0:N C1		C1	
1.3.3	retransmit	8.4.2.1 C3	F, T C1		F, T C1	
1.3.4	acknowledge	8.4.2.1 C4	F, T C1		F, T C1	
1.3.5	timeliness	8.4.2.1 C5	F, T C1		F, T C1	
1.3.6	modulus number of associated DLSDU or of highest-numbered DLSDU sent	8.4.2.1 C6	C1		C1	
1.3.7	(zero-origin) total number of segments in DLSDU	8.4.2.1 C7	0:N C1		C1	
1.3.8	(zero-origin) associated segment number	8.4.2.1 C8	0:N C1		C1	
1.4	format D — UNSEGMENTED LONG	8.4.2.1 D	C1		C1	
1.5	format E — SUBSCRIBER	8.4.2.1 E	C1		C1	
1.6	format F — PUBLISHER	8.4.2.1 F	C1		C1	
1.7	format G — UNSEGMENTED PUBLISHER	8.4.2.1 G	C1		C1	
C1 : **if** DLC_conformance_class = {A, B, or C} **then** M **else** O.						
N = : (the value claimed in E.10.2.2/2) minus 1						

E.10.2.6.3 Timeliness parameters

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status	Support	Status	Support
1	SD-parameter time-stamp format	8.4.2.2				
1.1	format J – no-time format	8.4.2.2 J	M		M	
1.2	format K – 2-second format	8.4.2.2 K	M		C1	
1.3	format L – 8-minute format	8.4.2.2 L	M		C1	
1.4	format M – full-time format	8.4.2.2 M	M		C1	
C1 **if** E.8.2.5/2 **and** E.8.2.6/2 **then** M **else** O						

E.10.3 Parameters of the SR DLPDU — protocol version 1

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status	Support	Status	Support
1	SR-parameter formats	8.5				
1.1	format X — NULL	8.5 a	M		C1	
1.2	format Y — ERROR	8.5 b	M		C1	
C1 **if** potential_LAS **then** M **else** O potential_LAS = E.9.1/2 or /3						

E.10.4 Parameters of the TD DLPDU — protocol version 1

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status	Support	Status	Support
1	link originating DL-time	8.6 a	M		C2	
2	DL-time-quality measures	8.6 b	M		C2	
2.1	limiting time-synchronism class	8.6 b1	0:7 M		0:N C2	
2.2	number of intervening links	8.6 b2	0:7 M		0:7 C2	
2.3	source type	8.6 b3	0:3 M		0:3 C2	
3	DL-time offset	8.6 c	C1		C2	
4	DL-time prior to end-of-transmission	8.6 d	C1		C2	
5	DL-time adjustment	8.6 e	C1		C2	

C1 **if** time_synchronism_supported **then** M **else** O time_synchronism_supported = E.8.3.1/2 **or** /3 **or** /4 **or** /5 **or** /6 **or** /7 **or** /8

C2 **if** potential_LAS **then** M **else** O potential_LAS = E.9.1/2 or /3

N = **if** E.8.3.1/1 **then** 0 **elseif** E.8.3.1/2 **then** 1 **elseif** E.8.3.1/3 **then** 2 **elseif** E.8.3.1/4 **then** 3 **elseif** E.8.3.1/5 **then** 4 **elseif** E.8.3.1/6 **then** 5 **elseif** E.8.3.1/7 **then** 6 **else** 7

E.10.5 Parameters of the RQ DLPDU — protocol version 1

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	Parameter	Reference	Status	Support	Status	Support
1	sending time-stamp	8.7 a	M		C1	

C1 **if** potential_LAS **then** M **else** O potential_LAS = E.9.1/2 or /3

E.10.6 Parameters of the RR DLPDU — protocol version 1

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status**	**Support**	**Status**	**Support**
1	DL-time-quality measures	8.8 a	C1		M	
1.1	limiting time-synchronism class	8.8.a1	0:7 M		0:N C2	
1.2	support for estimated C(NT)	8.8.a2	C2		M	
2	RQ sending time-stamp	8.8 a	C1		M	
3	RQ receiving time-stamp	8.8 a	C1		M	
4	RR sending time-stamp	8.8 b	C1		M	

C1 **if** potential_LAS **then** M **else** O   potential_LAS = E.9.1/2 or /3
C2 **if** potential_LAS **or** N¼0 **then** M **else** O
N = **if** E.8.3.1/1 **then** 0 **elseif** E.8.3.1/2 **then** 1 **elseif** E.8.3.1/3 **then** 2 **elseif** E.8.3.1/4 **then** 3

elseif E.8.3.1/5 **then** 4 **elseif** E.8.3.1/6 **then** 5 **elseif** E.8.3.1/7 **then** 6 **else** 7

E.10.7 Parameters of the PN DLPDU — protocol version 1

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status or Range**	**Support**	**Status or Range**	**Support**
1	DLC protocol version number	8.9 a3	1 M		1 M	
2	PhL inter-channel signal skew	8.9 a1	0:7 M		0:7 C1	
3	PhL post-transmission-gap extension units	8.9 a4	0:7 M		0:7 C1	
4	PhL preamble extension units	8.9 a5	0:7 M		0:7 C1	
5	slot-time	8.9 b1	1:4095 M		1:4095 C1	
6	maximum-response-delay	8.9 b2	1:12 M		1:12 C1	
7	minimum-inter-PDU-delay	8.9 b3	0:120 M		0:120 C1	

C1 **if** potential_LAS **then** M **else** O   potential_LAS = E.9.1/2 or /3

E.10.8 DD parameters of the PT, ES and RI DLPDUs — protocol version 1

Supported Parameters						
	Column letter →		A) Receiving and Recognizing		B) Forming and Sending	
Item number	**Parameter**	**Reference**	**Status**	**Support**	**Status**	**Support**
1	delegated or requested duration	8.10 a	0:65000 M		0:65000 M	

Annex F — (informative) Formal protocol finite state machines

(This annex does not form part of this standard.)

NOTE — A formal version of the finite state machine descriptions incorporated in this Annex, and of additional finite state machine descriptions, is under development. It is anticipated that when that formal version is available, it will be submitted as a Draft Addendum to replace this annex with a corresponding normative annex.

This clause specifies a number of finite state machines used by the DLE to provide its low-level and high-level protocol functions. This specification is complementary to, and subordinate to, the textual specification in the body of this Part of this International Standard.

The finite state machine descriptions given here are necessarily less than a complete description of an implementation. Additional requirements and considerations are found in the textual specification.

F.1 Basic reception and transmission FSMs

F.1.1 Nomenclature

$X \Leftarrow Y$	assignment of value of expression *Y* to name-expression *X*
$X \wedge Y$	logical AND of expressions *X* and *Y* with short-circuit evaluation (*Y* is evaluated only if *X* is TRUE)
$X \vee Y$	logical OR of expressions *X* and *Y* with short-circuit evaluation (*Y* is evaluated only if *X* is FALSE)
$!\ Y$	logical NOT of value of expression *Y*
$\exists(X,Y)$	THERE EXISTS AN *X* SUCH THAT *Y* — which is true if an object of the type of name-expression *X* satisfying Boolean condition *Y* exists, and false otherwise. If an object of the type of *X* satisfying *Y* exists, then one such object (usually the only one) is bound to the name-expression *X* for use throughout subsequent parts of the transition condition and the transition's actions.
$X \in \{\ Y \ldots\ \}$	CONTAINED IN — *X* is a member of the set of elements *Y* …
$X \notin \{\ Y \ldots\ \}$	NOT CONTAINED IN — *X* is not a member of the set of elements *Y* …
$X\,.\,Y$	field selection — the *Y* th component of name-expression *X*
$(X)\bullet$	pointer dereference — the object to which name-expression *X* points events, variables, procedures, and other FSM nomenclature
FCS_OK	FCS condition at RxA\ of just-received DLPDU; false if no DLPDU just received
C(CT)	8-bit down-counter, prescaled by V(ST), for LAS measurement of periods of bus inactivity**:** "Token Recovery" — preset with P(TRD), 15, or V(TN)
C(RR)	8-bit down-counter, prescaled by V(ST), for token-holder measurement of periods of bus inactivity**:** "Reply Recovery" — preset with V(MRD)+1
C(FD)	8-bit down-counter, prescaled by V(ST), for responder measurement of reply-forming delay**:** "Forming Delay" — preset with V(MRD)
V(TL)	16-bit link identifier**:** "This Link"
V(TN)	8-bit node identifier**:** "This Node"

V(DTA)	32-bit DL-address used to identify the current token holder: “Delegated Token Address”
V(DA)	32-bit DL-address used to retain the last DLPDU’s “Destination Address” if a CA, CD, ED or RQ DLPDU, else null (invalid)
V(SA)	32-bit DL-address used to retain the last DLPDU’s “Source Address” if a CA, CD, ED or RQ DLPDU, else null (invalid)
V(LN)	8-bit node identifier used to record the node-id of the current LAS: “LAS Node”
V(TT)	8-bit FC as “Token Type:” TL, ES, PT, CA, CD, ED

NOTE — CA, CD, ED are considered to be half-transaction tokens.

V(NTS)	16-bit node-time source link number
V(NTO)	56-bit DL-time offset with respect to sending DLE’s node-time
C(NT)	56-bit frequency-corrected up-counter: “Node Time”
C(UC)	16-bit down-counter: “Unscheduled Capacity” used by LAS to generate ES and PT DLPDUs
C(RD)	16-bit down-counter: “Remaining Duration” used by token-holder to limit use of ES-token
Q(LR)	prioritized FIFO queue of unscheduled SPDUs for transmission to the local link’s LAS
Q_N(US)	prioritized FIFO queue of unscheduled service requests for the DLE
Q_X.is_empty	Boolean indicating that the specified queue is or is not empty
Q_X(Y,min=PP)	subset of Q_X(Y) whose priority is PP or higher
rcv	record holding parsed version of just-received DLPDU
next	record holding parsed version of DLPDU in the selected transmit scheduling queue
reply	record holding constructed immediate reply parameters
xxx.yyy	field yyy of record xxx
xxx.FC.F	Boolean field of FC field of xxx DLPDU which indicates final transaction of token use
xxx.FC.DA	DA field is present in xxx DLPDU — determination based on FC value
xxx.FC.SA	SA field is present in xxx DLPDU — determination based on FC value
xxx.DA.exists	record xxx contains a DA field
xxx.SA.exists	record xxx contains an SA field
xxx.is_group	variable xxx specifies a group DL-address
xxx.is_DLSAP_addr	variable xxx specifies a DLSAP-address
xxx.is_DLCEP_id	variable xxx specifies the DL-identifier of a DLCEP
is_usable(xxx)	the implementation is capable of operation with the value of parameter xxx
falling_edge(xxx)	event detecting falling (TRUE → FALSE) edge of signal xxx
dlm_queue_deliver	report received DLPDU to DL-management for processing
dlm_event	notify DL-management of specified event
process_xxx	process specified type of received DLPDU

send_xxx(…) send specified DLPDU with specified explicit arguments and appropriate values for other arguments

send_item send specified DLPDU from parametric description, providing current values for the C(NT)-dependent arguments on TD, RQ and RR DLPDUs

local_addr DLE’s address-recognition function — recognizes all but multi-peer subscriber DLCEPs

subscriber DLE’s other address-recognition function — recognizes only multi-peer subscriber DLCEPs

create_object(*X*) create an object of the type of *X* and bind it to the name local name *X*

delete_object delete specified object

activate_addr_recogition activate recognition of the specified address

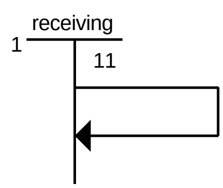
local_object local object addressed by specified DL-address, for which local_addr is true

FSM_state(FSM_name, FSM_state_number Boolean condition, true if specified FSM is in specified state

FSM_transition(FSM_name, transition_state_number_pair event, true when specified FSM executes the specified transition (from the first to the second specified state)

F.1.2 The receiver FSM

receiver FSM



receiver FSM

state	name	transitions	entry actions
1	receiving	11	

transition condition **:** [actions]

11a rcv.FC=EC ∧ ((rcv.FC.DA ∧ local_addr(rcv.DA)
∨ (!rcv.FC.DA ∧ subscriber(rcv.SA))
: process_EC(rcv);
V(DA) ⇐ null;
V(SA) ⇐ null;

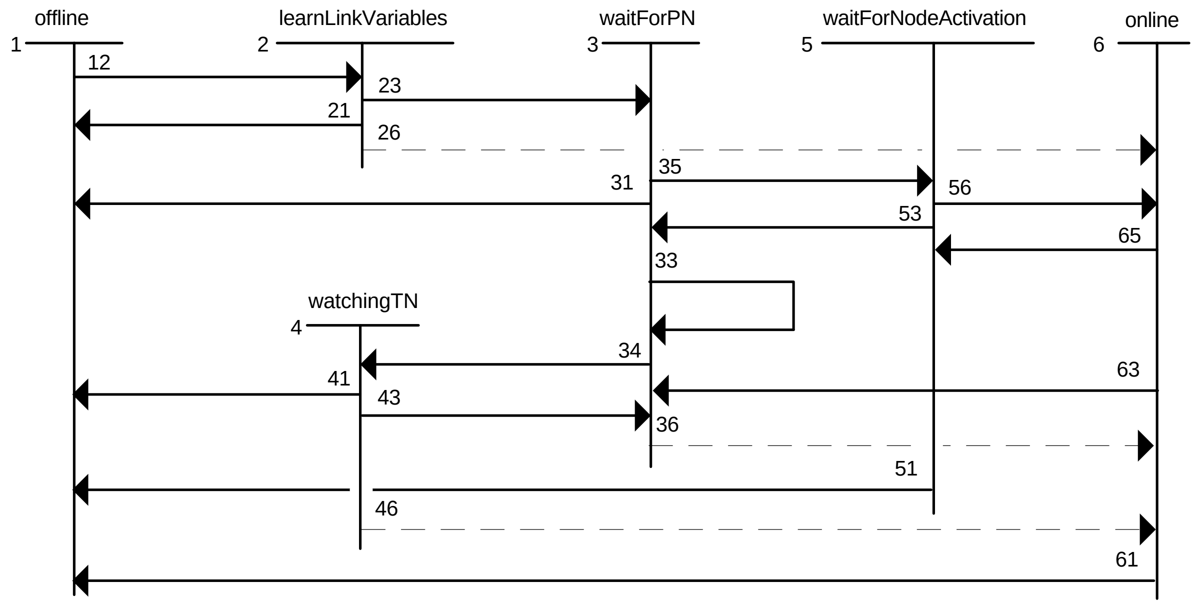
11b rcv.FC=DC ∧ ((rcv.FC.DA ∧ local_addr(rcv.DA)
∨ (!rcv.FC.DA ∧ subscriber(rcv.SA))
: process_DC(rcv);
V(DA) ⇐ null;
V(SA) ⇐ null;

11c rcv.FC=RC ∧ ((rcv.FC.DA ∧ local_addr(rcv.DA)
∨ (!rcv.FC.DA ∧ subscriber(rcv.SA))

```
                :   process_RC(rcv);
                    V(DA) ⇐ null;
                    V(SA) ⇐ null;
11d         rcv.FC=CA ∧ local_addr(rcv.DA)
                :   process_CA(rcv);
                    V(DA) ⇐ rcv.DA;
                    V(SA) ⇐ rcv.SA;
11e         rcv.FC=CD ∧ local_addr(rcv.DA)
                :   process_CD(rcv);
                    V(DA) ⇐ rcv.DA;
                    V(SA) ⇐ rcv.SA;
11f         rcv.FC=ED ∧ local_addr(rcv.DA)
                :   process_ED(rcv);
                    V(DA) ⇐ rcv.DA;
                    V(SA) ⇐ rcv.SA;
11g         rcv.FC ⇐ DT ∧ (  (  rcv.FC.DA ∧ local_addr(rcv.DA))
                          ∨ ( !rcv.FC.DA ∧  rcv.FC.SA ∧ subscriber(rcv.SA))
                          ∨ ( !rcv.FC.DA ∧ !rcv.FC.SA ∧ subscriber(V(DA)))
                          ∨ ( !rcv.FC.DA ∧ !rcv.FC.SA ∧ local_addr(V(SA))) )
                :   process_DT(rcv);
                    V(DA) ⇐ null;
                    V(SA) ⇐ null;
11h         rcv.FC=TD
                :   rcv.time_of_receipt ⇐ C(NT);
                    V(LN) ⇐ rcv.SA.N;/* track LAS node-id */
                    process_TD(rcv);
                    V(DA) ⇐ null;
                    V(SA) ⇐ null;
11j         rcv.FC ∈ {CL,TL}
                :   V(LN) ⇐ rcv.SA.N;/* track LAS node-id */
                    dlm_event(CL_or_TL_received);
                    V(DA) ⇐ null;
                    V(SA) ⇐ null;
11k         rcv.FC=WK ∧ rcv.DA.N=V(TN)
                :   power_up_local_circuitry;
                    dlm_event(WK_received);
                    V(DA) ⇐ null;
                    V(SA) ⇐ null;
```

F.1.3 The DLE-state FSM

DLE-state FSM



responding FSM

state	name	transitions	entry actions
1	offline	12	
2	learnLinkVars	21, 23	
3	waitForPN	31, 33, 34, 35	**if** V(TN)³x'F8' **then** $V(TN)_{1-0}$ Ü (random Ù x'03');
4	watchingTN	41, 43	C(RR) Ü V(MRD)+1;
5	waitForNodeActivation	51, 53, 56	V(RID) Ü random; send PR(V(RID));
6	online	61, 65	

transition condition **:** [actions]

1 DLM_action("go online")

:

***n*1** DLM_action("go offline")

(2≤*n*≤6) **:**

23 rcv.FC=PN

: set essential PhL and DLL variables from PN-parameters;

33a rcv.FC=PT ∧ rcv.DA.N=V(TN) ∧ V(TN) ≥ x'F8'

:

33b rcv.FC=PN ∧ rcv.DA.N=V(TN) ∧ !(is_usable(V(MID)) ∧ is_usable(V(MRD)))

:

: V(RID) ⇐ random;

send PR(V(RID));

34 rcv.FC=PT ∧ rcv.DA.N=V(TN) ∧ V(TN) < x'F8'

:

35 rcv.FC=PN ∧ rcv.DA.N=V(TN) ∧ is_usable(V(MID)) ∧ is_usable(V(MRD))

:

43a rcv.FC=any

: V(TN) ⇐ x'F8';

43b C(RR)=0

:

53a rcv.FC ∈ {PN,PT} ∧ rcv.DA.N=V(TN)

:

53b rcv.FC=DT ∧ rcv.DA.N=V(TN) ∧ rcv.DA.S=00

∧ (rcv.data.SPDU_type≠"node activation"

∨ rcv.data.node_activation_SPDU.N(RID)≠V(RID))

:

56b rcv.FC=DT ∧ rcv.DA.N=V(TN) ∧ rcv.DA.S=00

∧ rcv.data.SPDU_type="node activation"

∧ rcv.data.node_activation_SPDU.N(RID)=V(RID))

: set remaining basic DLL variables from SPDU;

63 rcv.FC=PN ∧ rcv.DA.N=V(TN) ∧ !(is_usable(V(MID)) ∧ is_usable(V(MRD)))

:

65 rcv.FC=PN ∧ rcv.DA.N=V(TN) ∧ is_usable(V(MID)) ∧ is_usable(V(MRD))

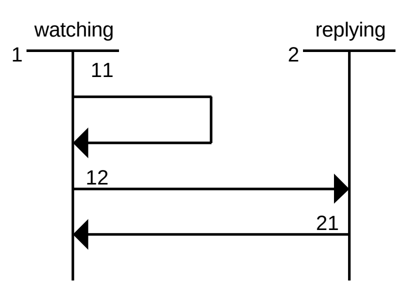
:

n6a DLE_is_LM ∧ DLE_knows_DLL_variables ∧ FSM_transition(link_master_FSM, 12)

(2≤***n***≤5) :

F.1.4 The responder FSM

responding FSM



responding FSM

state	name	transitions	entry actions
1	watching	11, 12	
2	replying	21	C(FD) Ü V(MRD)

transition condition : [actions]

11a rcv.FC∉ {CA,CD,ED,RQ} ∨ rcv.DA.is_group ∨ !local_addr(rcv.DA)

$\vee$ rcv_data_length$\neq$0

:

11b rcv.FC $\in$ {CA,CD,ED} $\wedge$!rcv.DA.is_group $\wedge$ local_addr(rcv.DA)

$\wedge$ rcv_data_length=0

$\wedge$!(local_object(rcv.DA).typ $\in$ {peer_DLCEP,publisher_DLCEP}

$\wedge$ (!rcv.FC.SA $\vee$ rcv.SA=local_object(rcv.DA).remote_DLCEP_addr))

: send_SR(V(TN));

12a rcv.FC $\in$ {CA,CD,ED} $\wedge$!rcv.DA.is_group $\wedge$ local_addr(rcv.DA)

$\wedge$ (local_object(rcv.DA).typ $\in$ {peer_DLCEP,publisher_DLCEP}

$\wedge$ (!rcv.FC.SA $\vee$ rcv.SA=local_object(rcv.DA).remote_DLCEP_addr))

: form_DT_reply per rcv.DA;

12b rcv.FC=RR $\wedge$ local_addr(rcv.DA)

: form_RR_reply;

21a reply_formed

: send_item(next);

21b C(FD)=0

:

form_DT_reply per rcv.DA

{ next $\Leftarrow$ DT_reply_to(rcv.DA);

if Q_W(SS).is_empty $\wedge$!Q(LR).is_empty

then move first element of Q(LR) to end of appropriate priority of Q_W(SS);

if !Q_W(SS).is_empty

then next.FC.DD $\Leftarrow$ priority(Q_W(SS).head);

}

form_RR_reply

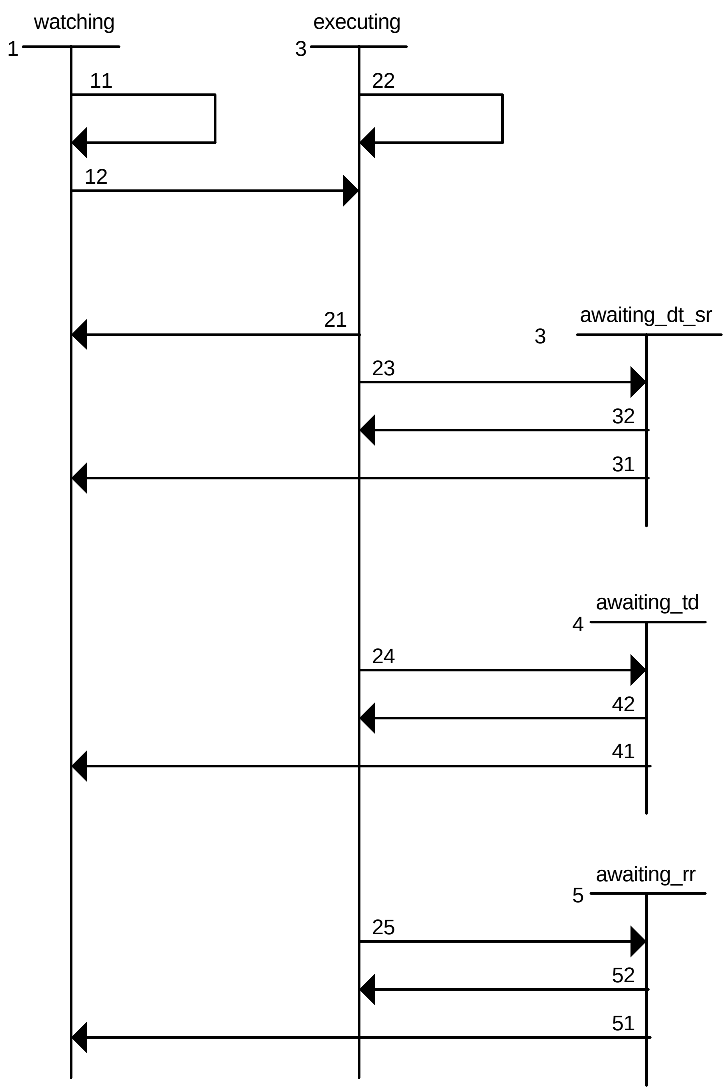
{ next $\Leftarrow$ RR_reply_to(rcv.DA);

add value of C(NT) at the time of sending;

}

F.1.5 The token-holder FSM

token holding FSM



token holding FSM

state	name	transitions	entry actions
1	watching	12	
2	executing	21, 22, 23, 24, 25	select_next
3	awaiting_dt_sr	31, 32	C(RR) Ü V(MRD)+1
4	awaiting_td	41, 42	C(RR) Ü V(MRD)+1
5	awaiting_rr	51, 52	C(RR) Ü V(MRD)+1

<u>transition</u> <u>condition **:** [actions]</u>

12a rcv.FC=PT Ù rcv.DA.N = TN
: V(TT) Ü rcv.FC;
C(RD) Ü rcv.PT_duration;
next.FC.F Ü NOT-FINAL;

12b rcv.FC=ES Ù !rcv.DA.is_group Ù local_addr(rcv.DA) Ù
local_object(rcv.DA).typ = DLSEP
: V(TT) Ü rcv.FC;
C(RD) Ü rcv.ES_duration;
next.FC.F Ü NOT-FINAL;

21 next.FC **Î**{RI,RT} Ú (next.FC **Î**{EC,DC,RC,DT,IDLE} Ù next.FC.F)
: send_item(next);

22 next.FC **Î**{EC,DC,RC,DT,IDLE} Ù !next.FC.F
: send_item(next);

23 next.FC **Î**{CA,CD,ED}
: send_item(next);
V(DA) Ü next.DA;
V(SA) Ü next.SA;

24 next.FC=CT
: send_item(next);

25 next.FC=RQ
: send_item(next);

31 !(rcv.FC=SR Ú (rcv.FC **Î**{DT} Ù !next.FC.F Ù rcv.FC.DA Ù rcv.DA=V(SA)))
:

32a rcv.FC=SR Ú (rcv.FC **Î**{DT} Ù !next.FC.F Ù rcv.FC.DA Ù rcv.DA=V(SA))
:

32b C(RR)=0
:

41 rcv.FC¹TD
: abort_token=TRUE;

42a rcv.FC=TD
:

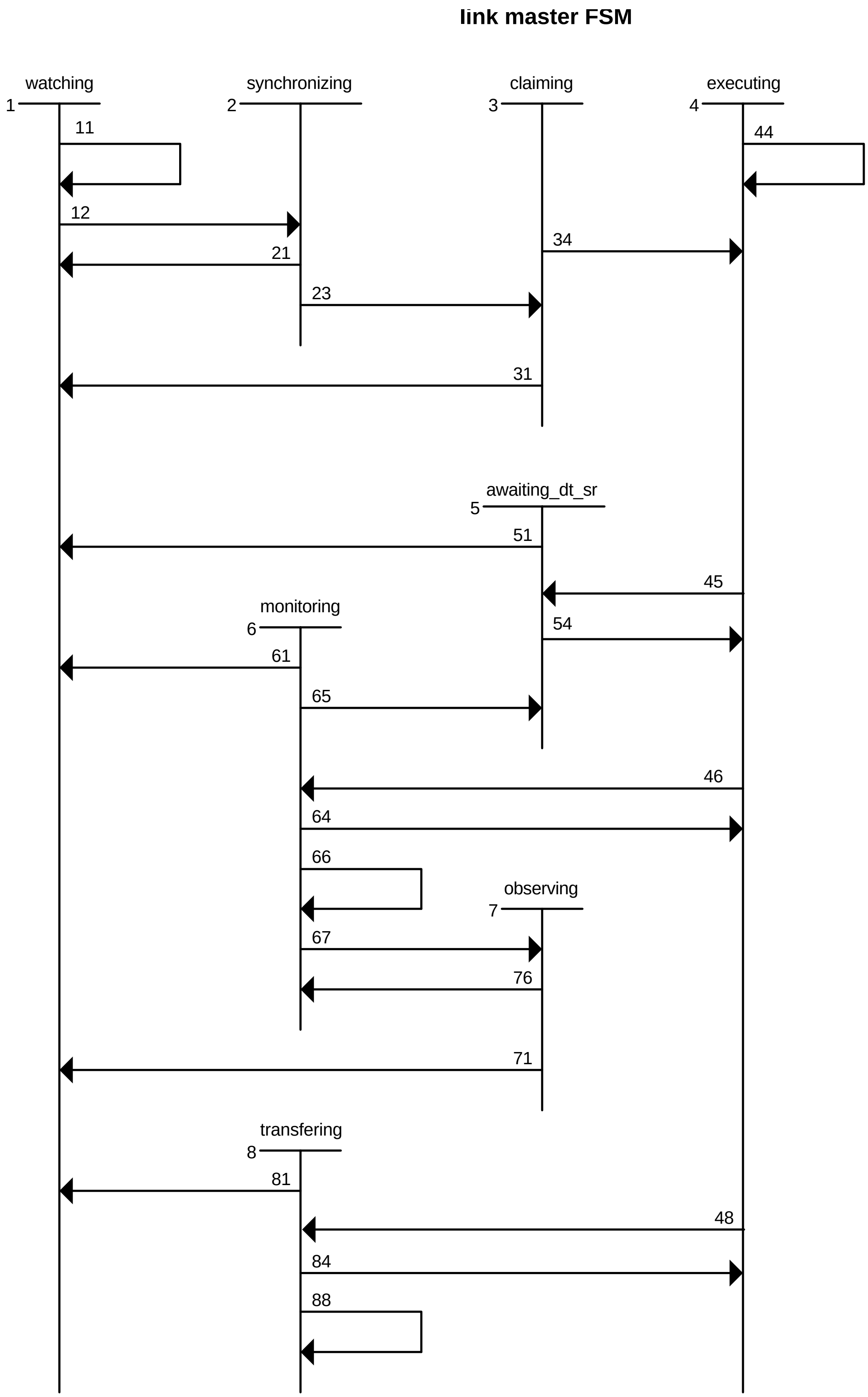
42b C(RR)=0
:

51 rcv.FC¹RR Ú rcv.DA¹V(SA)
: abort_token=TRUE;

52a rcv.FC=RR Ù rcv.DA=V(SA)
:

52b C(RR)=0
:

F.1.6 The link-master FSM



LM FSM

state	name	transitions	entry actions
1	watching	11, 12	C(CT) Ü V(TN)
2	synchronizing	21, 23	C(CT) Ü V(TN)
3	claiming	31, 34	C(CT) Ü random Ù x'03'
4	executing	44, 45, 46, 48	select next per schedule
5	awaiting_dt_sr	51, 54	C(CT) Ü V(MRD)+1
6	monitoring	61, 64, 65, 66, 67	C(CT) Ü P(TRD)
7	observing	71, 76	C(CT) Ü P(TRD)
8	transferring	81, 84, 88	C(CT) Ü x'0F'

transition	condition : [actions]
11	falling_edge(RxA) :
12	C(CT)=0 Ù !FSM_state(link_master_FSM, 1) : send_CL(V(TN));
21	falling_edge(RxA) :
23	C(CT)=0 : send_CL(V(TN));
31	falling_edge(RxA) :
34	C(CT)=0 :
44	next.FC Î{EC,DC,RC,DT,TD,WK,IDLE} : send_item(next);
45	next.FC Î{CA,CD,ED} : send_item(next); V(TT) Ü next.FC; V(DA) Ü next.DA;
46a	next.FC=PT : send_item(next); V(TT) Ü next.FC; V(DTA) Ü next.DA;
46b	next.FC=ES Ù next.ES_duration¹0 : send_item(next); V(TT) Ü next.FC; V(DTA) Ü next.DA;
46c	next.FC=ES Ù next.ES_duration=0 : next.ES_duration Ü C(UC);/* within the LAS' schedule an ES_duration */ send_item(next); /* of 0 requests a pseudo-circulated token */ V(TT) Ü next.FC; V(DTA) Ü next.DA;
48	next.FC=TL : send_item(next); V(TT) Ü next.FC; V(DTA) Ü next.DA;

51 falling_edge(RxA)
Ù !(rcv.FC=SR Ú (rcv.FC=DT Ù !rcv.FC.F Ù (!rcv.FC.SA Ú rcv.SA=V(DA))))
:

54 rcv.FC=SR Ú (rcv.FC=DT Ù !rcv.FC.F Ù (!rcv.FC.SA Ú rcv.SA=V(DA)))
:

61 rcv.FC **Î**{WK,TD}
:

64 rcv.FC **Î**{EC,DC,RC,DT} Ù rcv.FC.F
: V(DA) Ü rcv.DA;

65 rcv.FC **Î**{CA,CD,ED} Ù rcv.FC.F
: V(DA) Ü rcv.DA;

66a rcv.FC **Î**{EC,DC,RC,DT} Ù !rcv.FC.F
:

66b rcv.FC **Î**{IDLE}
:

66c rcv.FC=CT
: send_TD(V(NTS),V(NTQ),V(NTO),C(NT));

66d C(CT)=0
: send_RT(V(TN));

67 rcv.FC **Î**{CA,CD,ED,RQ} Ù !rcv.FC.F
: V(DA) Ü rcv.DA;

71 falling_edge(RxA)
Ù !(rcv.FC **Î**{SR,RR}
Ú (rcv.FC=DT Ù !rcv.FC.F Ù (!rcv.FC.SA Ú rcv.SA=V(DA))))
:

76a rcv.FC **Î**{SR,RR} Ú (rcv.FC=DT Ù !rcv.FC.F Ù (!rcv.FC.SA Ú rcv.SA=V(DA)))
:

76b C(CT)=0
:

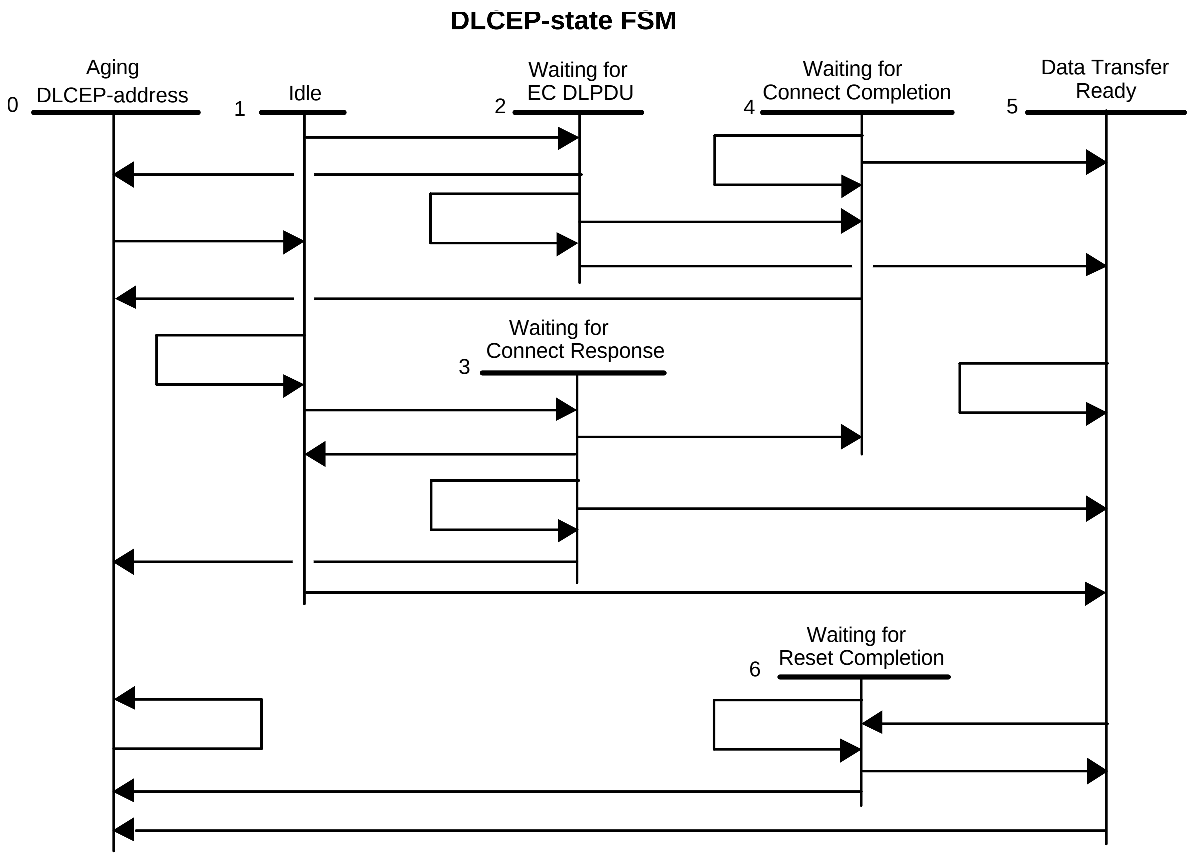
81 falling_edge(RxA) Ù FCS_OK /* This is equivalent to rcv.FC in all */
:

84 C(CT)=0
:

88 falling_edge(RxA) Ù !FCS_OK
:

F.2 FSMs for DLCs

F.2.1 The DLCEP-state FSM



DLCEP-state FSM

state	name	transitions
0	agingDLCEPaddress	00, 01
1	idle	11, 12, 13, 15
2	waitingForEC_DLPDU	20, 22, 24, 25
3	waitingForConnectResponse	30, 31, 33, 34, 35
4	waitingForConnectCompletion	40, 44, 45
5	dataTransferReady	50, 55, 56
6	waitingForResetCompletion	60, 65, 66

NOTE — Within states 2 through 6, non-hierarchical variable references are references to variables of the active DLCEP.

transition	condition : [actions]
01	timeout : delete_object(DLCEP);
11a	DL-CONNECT.req ∧ ((calling_id=DLCEP_id ∧ class≠PUBLISHER) ∨ !accepted_by_DLE) : DL-DISCONNECT.ind;
11b	rcv.FC=EC ∧ !accepted_by_DLE ∧ rcv.DA_exists ∧ !rcv.DA.is_group

: send_DC(rcv.SA, rcv.DA, !reply_requested);

11c rcv.FC=EC ∧ !accepted_by_DLE ∧ (!rcv.DA_exists ∨ rcv.DA.is_group)

:

11d rcv.FC=EC ∧ rcv.DA_exists ∧ (rcv.DA.is_DLSAP_addr ∨ rcv.DA.is_group)

∧ ∃(DLCEP, DLCEP.remote_DLCEP_addr=rcv.SA)

∧ DLCEP.remote_DLSAP_addr≠rcv.SA2

: send_DC(rcv.SA, rcv.DA, !reply_requested);

11e rcv.FC=EC ∧ rcv.DA_exists ∧ (rcv.DA.is_DLSAP_addr ∨ rcv.DA.is_group)

∧ ∃(DLCEP, DLCEP.remote_DLCEP_addr=rcv.SA)

∧ DLCEP.remote_DLSAP_addr=rcv.SA2

:

12 DL-CONNECT.req ∧ accepted_by_DLE ∧ class≠PUBLISHER

: create_object(DLCEP);

DLCEP.DLCEP_addr ⇐ requested_or_new_DLCEP(req);

DLCEP.DLSAP_addr ⇐ req.calling_DLSAP_addr;

DLCEP.called_addr ⇐ req.called_addr;

DLCEP.required_interactions ⇐ cnf;

activate_addr_recogition(DLCEP.DLCEP_addr);

if DLCEP.called_addr≠UNKNOWN

then send_EC(DLCEP.called_addr, DLCEP.DLCEP_addr,

DLCEP.DLSAP_addr, reply_requested);

start_timer(DLCEP.send_params.max_confirm_delay_for_EC

/(V(MRC)+1));

13 rcv.FC=EC ∧ rcv.DA_exists ∧ (rcv.DA.is_DLSAP_addr ∨ rcv.DA.is_group)

∧ ! ∃(DLCEP, DLCEP.remote_DLCEP_addr=rcv.SA)

: create_object(DLCEP);

DLCEP.remote_DLCEP_addr ⇐ rcv.SA;

DLCEP.remote_DLSAP_addr ⇐ rcv.SA2;

DLCEP.required_interactions ⇐ rsp_ind;

DL-CONNECT.ind(DLCEP);

start_timer(DLCEP, rcv.EC_p.send_params.max_confirm_delay_for_EC);

--

-- The following assignment is needed temporarily, until a DL-CONNECT.rsp is

-- received, for use with any requested or internally-generated DC DLPDU.

-- It will not be used if rcv.DA was a group DL-address.

--

DLCEP.DLCEP_addr ⇐ rcv.DA;

15 DL-CONNECT.req ∧ accepted_by_DLE ∧ class=PUBLISHER

:if !req.DLCEP.calling_id.is_DLCEP_id

then create_object(DLCEP);

DLCEP.DLCEP_addr ⇐ requested_or_new_DLCEP(req);

activate_addr_recogition(DLCEP.DLCEP_addr);

DLCEP.DLSAP_addr ⇐ req.calling_DLSAP_addr;

DLCEP.called_addr ⇐ req.called_addr;

if req.called_addr=UNKNOWN

then send_EC(—, DLCEP.DLCEP_addr, DLCEP.calling_addr,

!reply_requested);

else send_EC(DLCEP.called_addr, DLCEP.DLCEP_addr,

DLCEP.DLSAP_addr, !reply_requested);

DL-CONNECT.cnf;

20b timeout ∧ (retry_count=max_retry_count) ∧ !called_addr.is_group

:{see generic ***n***0b transition}

20c timeout ∧ (retry_count=max_retry_count) ∧ called_addr.is_group

:{see generic ***n***0c transition}

20e rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ !accepted_by_DLE

∧ !called_addr.is_group

: DL-DISCONNECT.ind

send_DC(rcv.SA, DLCEP_addr, !reply_requested);

start_timer();

22a timeout ∧ (retry_count < max_retry_count)

: send_EC(called_addr, DLCEP_addr, DLSAP_addr, reply_requested);

start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

22b rcv.FC=EC ∧ !accepted_by_DLE ∧ (!rcv.DA_exists ∨ called_addr.is_group)

: send_DC(rcv.SA, DLCEP_addr, !reply_requested);

24a rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ accepted_by_DLE

∧ rcv.EC_p.reply_requested ∧ class=PEER

: remote_DLCEP_addr ⇐ rcv.SA;

remote_DLSAP_addr ⇐ rcv.SA2;

start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

send_EC(remote_DLCEP_addr, DLCEP_addr, DLSAP_addr,

!reply_requested);

24b merge_from_another_DLCEP

: start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

send_EC(remote_DLCEP_addr, DLCEP_addr, DLSAP_addr,

!reply_requested);

25a rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ accepted_by_DLE

∧ (class=SUBSCRIBER ∨ !rcv.EC_p.reply_requested)

: remote_DLCEP_addr ⇐ rcv.SA;

remote_DLSAP_addr ⇐ rcv.SA2;

stop_timer;

if class=PEER

then if SA_is_required

then send_DT(rcv.SA, DLCEP_addr);

else send_DT(rcv.SA, —);

DL-CONNECT.cnf

25b rcv.FC=EC ∧ !rcv.DA_exists ∧ rcv.SA=DLCEP_addr ∧ class=SUBSCRIBER

∧ accepted_by_DLE

: remote_DLCEP_addr ⇐ rcv.SA;

remote_DLSAP_addr ⇐ rcv.SA2;

reset_seq_numbers_to_next_received ⇐ TRUE;

DL-CONNECT.cnf

30b (timeout ∨ (DL-CONNECT.rsp ∧ !accepted_by_DLE)) ∧ !called_addr.is_group

∧ class≠SUBSCRIBER

: see generic ***n***0b transition}

30c (timeout ∨ (DL-CONNECT.rsp ∧ !accepted_by_DLE))

∧ (called_addr.is_group ∨ class=SUBSCRIBER)

: {see generic ***n***0c transition}

31a DL-CONNECT.rsp ∧ responding_id.is_DLCEP_id ∧ accepted_by_DLE
∧ responding_id.class=PEER
: (DLCEP.responding_addr)•.remote_DLCEP_addr
⇐ DLCEP.remote_DLCEP_addr;
(DLCEP.responding_addr)•.remote_DLSAP_addr
⇐ DLCEP.remote_DLSAP_addr;
signal (DLCEP.responding_addr)•.DLCEP(merge_from_another_DLCEP);
delete_object(DLCEP);

31b DL-CONNECT.rsp ∧ responding_id.is_DLCEP_id ∧ accepted_by_DLE
∧ responding_id.class=PUBLISHER
:send_EC(remote_DLCEP_addr, (DLCEP.responding_addr)•.DLCEP_addr,
(DLCEP.responding_addr)•.DLSAP_addr, !reply_requested);
delete_object(DLCEP);

33b rcv.FC=EC ∧ !rcv.DA_exists ∧ rcv.SA_exists ∧ rcv.SA=remote_DLCEP_addr
∧ (rcv.publisher-DLCEP-address-reuse-discriminator
=publisher-DLCEP-address reuse-discriminator)
:

34 DL-CONNECT.rsp ∧ !responding_id.is_DLCEP_id ∧ accepted_by_DLE
∧ class=PEER
: DLCEP_addr ⇐ new_or_requested_DLCEP_addr(rsp);
DLSAP_addr ⇐ rsp.calling_DLSAP_addr;
activate_addr_recogition(DLCEP_addr);
send_EC(remote_DLCEP_addr, DLCEP_addr, DLSAP_addr,
!reply_requested);
start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

35 DL-CONNECT.rsp ∧ !responding_id.is_DLCEP_id ∧ accepted_by_DLE
∧ class≠PEER
: DLCEP_addr ⇐ new_or_requested_DLCEP_addr(rsp);
DLSAP_addr ⇐ rsp.calling_DLSAP_addr;
stop_timer;
activate_addr_recogition(DLCEP_addr);
if class=PUBLISHER
then send_EC(remote_DLCEP_addr, DLCEP_addr, DLSAP_addr,
!reply_requested);
elsereset_seq_numbers_to_next_received ⇐ TRUE;
DL-CONNECTION-ESTABLISHED.ind;

40b timeout
: {see generic ***n***0b transition}

44b rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.EC_p.reply_requested
∧ (rcv.DA=DLCEP_addr
∨ ((rcv.DA.is_DLSAP_addr ∨ rcv.DA.is_group)
∧ ∃(DLCEP, DLCEP.remote_DLCEP_addr=rcv.SA)
∧ DLCEP.remote_DLSAP_addr=rcv.SA2))
: send_EC(rcv.SA, DLCEP_addr, DLSAP_addr, !reply_requested);
start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

44c timeout ∧ (retry_count < max_retry_count)
: send_EC(called_addr, DLCEP_addr, DLSAP_addr, reply_requested);
start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

44d rcv.FC=EC ∧ !rcv.DA_exists ∧ rcv.SA_exists ∧ rcv.SA=remote_DLCEP_addr
∧ (rcv.publisher-DLCEP-address-reuse-discriminator
=publisher-DLCEP-address reuse-discriminator)
:

45a rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ !rcv.EC_p.reply_requested
: stop_timer;
if SA_is_required
then send_DT(rcv.SA, DLCEP_addr);
else send_DT(rcv.SA, —);
if required_interactions=cnf
then DL-CONNECT.cnf;
else DL-CONNECTION-ESTABLISHED.ind;

45b rcv.FC=DT
: stop_timer;
if required_interactions=cnf
then DL-CONNECT.cnf;
else DL-CONNECTION-ESTABLISHED.ind;

50c rcv.FC=EC ∧ !rcv.DA_exists ∧ rcv.SA_exists ∧ rcv.SA=remote_DLCEP_addr
∧ (rcv.publisher-DLCEP-address-reuse-discriminator
≠publisher-DLCEP-address reuse-discriminator)
: DL-DISCONNECT.ind;
start_timer();

55b rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ class=PUBLISHER
∧ accepted_by_DLE ∧ rcv.EC_p.reply_requested
: send_EC(—, DLCEP_addr, DLSAP_addr, !reply_requested);

55c rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr
∧ class=PUBLISHER ∧ !(accepted_by_DLE ∧ rcv.EC_p.reply_requested)
: send_DC(rcv.SA, rcv.DA, !reply_requested);

55d rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr
∧ class≠PUBLISHER
:

55e rcv.FC=EC ∧ !rcv.DA_exists ∧ rcv.SA_exists ∧ rcv.SA=remote_DLCEP_addr
∧ (rcv.publisher-DLCEP-address-reuse-discriminator
=publisher-DLCEP-address reuse-discriminator)
:

55f rcv.FC ∈ {CA,CD,ED} ∧rcv.DA=DLCEP_addr
∧ ((!rcv.SA_exists ∧ !SA_is_required) ∨ rcv.SA=remote_DLCEP_addr)
: …

55g rcv.FC=DT ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr
∧ ((!rcv.SA_exists ∧ !SA_is_required) ∨ rcv.SA=remote_DLCEP_addr)
: …

55h rcv.FC=DT ∧ !rcv.DA_exists
∧ ((rcv.SA_exists ∧ rcv.SA=remote_DLCEP_addr)
∨ (rcv.SA_was_implied ∧ !SA_is_required
∧ rcv.implied_SA=remote_DLCEP_addr))
:…

55j DL-DATA.req ∨ DL-SUBSCRIBER-QUERY.req

:…

55k timeout ∧ (retry_count < max_retry_count)

:…

55m rcv.FC=RC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ class=PUBLISHER

: send_DC(rcv.SA, rcv.DA, !reply_requested);

56a rcv.FC=RC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ class=PEER

∧ rcv.SA=remote_DLCEP_addr

: DL-RESET.ind;

flush_internal_queues;

required_interactions ⇐ rsp_ind;

pended_DLPDUs ⇐ RC;

expected_DLPDUs ⇐ DT;

rcv_params.NDS ⇐ rcv.RC_p.NDS;

rcv_params.NDR ⇐ rcv.RC_p.NDR;

start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

56b rcv.FC=RC ∧ !rcv.DA_exists ∧ rcv.SA=DLCEP_addr ∧ class=SUBSCRIBER

: DL-RESET.ind;

flush_internal_queues;

required_interactions ⇐ rsp_ind;

pended_DLPDUs ⇐ none;

expected_DLPDUs ⇐ none;

56c DL-RESET.req

: flush_internal_queues;

required_interactions ⇐ cnf;

pended_DLPDUs ⇐ none;

if class=PEER

then expected_DLPDUs ⇐ RC;

send_RC(remote_DLCEP_addr, DLCEP_addr, reply_requested,

V_C(M), V_C(L)+1);

start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

elseif class=PUBLISHER

then expected_DLPDUs Ü none;

send_RC(—, DLCEP_addr, !reply_requested, V_C(N));

else expected_DLPDUs ⇐ none;

56d internal reset condition

: DL-RESET.ind;

flush_internal_queues;

required_interactions ⇐ rsp_ind;

pended_DLPDUs ⇐ none;

if class=PEER

then expected_DLPDUs ⇐ RC;

send_RC(remote_DLCEP_addr, DLCEP_addr, reply_requested,

V_C(M), V_C(L)+1);

start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

elseif class=PUBLISHER

then expected_DLPDUs ⇐ none;

send_RC(—, DLCEP_addr, !reply_requested, V_C(N))

else expected_DLPDUs ⇐ none;

60b timeout ∧ (required_interactions=rsp

∨ (expected_DLPDUs≠none ∧ retry_count=max_retry_count))

:{ see generic ***n***0b transition}

65 required_interactions≠rsp_ind ∧ expected_DLPDUs=none

: reset_seq_numbers_to_next_received ⇐ (class=SUBSCRIBER);

flush_internal_queues;

if required_interactions=cnf

then DL-RESET.cnf;

else DL-RESET-COMPLETED.ind;

66b rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ class=PUBLISHER

∧ rcv.EC_p.reply_requested

: expected_DLPDUs ⇐ none;

send_EC(rcv.SA, DLCEP_addr, DLSAP_addr, !reply_requested);

66c rcv.FC=EC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr

∧ (class≠PUBLISHER ∨ !rcv.EC_p.reply_requested)

:

66d rcv.FC=EC ∧ !rcv.DA_exists ∧ rcv.SA_exists ∧ rcv.SA=remote_DLCEP_addr

:

66e rcv.FC=RC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ class=PEER

∧ rcv.SA=remote_DLCEP_addr ∧ required_interactions≠rsp_ind

∧ rcv.RC_p.reply_requested

: **if** required_interactions=cnf

then expected_DLPDUs ⇐ RC;

else expected_DLPDUs ⇐ DT;

send_RC(rcv.SA, DLCEP_addr, !reply_requested, V_C(M), V_C(L)+1);

start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

66f rcv.FC=RC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ class=PEER

∧ rcv.SA=remote_DLCEP_addr ∧ required_interactions≠rsp_ind

∧ expected_DLPDUs=RC ∧ !rcv.RC_p.reply_requested

: expected_DLPDUs ⇐ none;

if SA_is_required

then send_DT(rcv.SA, DLCEP_addr);

else send_DT(rcv.SA, —);

66g rcv.FC=RC ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ class=PEER

∧ rcv.SA=remote_DLCEP_addr ∧ required_interactions=rsp_ind

: **if** rcv.RC_p.reply_requested

thenpended_DLPDUs ⇐ RC;

else pended_DLPDUs ⇐ DT;

66h rcv.FC=RC ∧ !rcv.DA_exists ∧ rcv.SA=DLCEP_addr ∧ class=SUBSCRIBER

:

66j rcv.FC=DT ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr ∧ class=PEER

∧ ((!rcv.SA_exists ∧ !SA_is_required) ∨ rcv.SA=remote_DLCEP_addr)

∧ required_interactions≠rsp_ind ∧ expected_DLPDUs=DT;

∧ (rcv.SD_p.NDS ≥ rcv_params.NDS)

: expected_DLPDUs ⇐ none;

66k DL-RESET.rsp ∧ required_interactions=rsp_ind

: required_interactions=ind;

if pended_DLPDUs=RC

then expected_DLPDUs ⇐ DT;

send_RC(rcv.SA, DLCEP_addr, !reply_requested, V_C(M), V_C(L)+1);

start_timer(send_params.max_confirm_delay_for_EC/(V(MRC)+1));

elseif pended_DLPDUs=DT

then expected_DLPDUs ⇐ none;

if SA_is_required

then send_DT(rcv.SA, DLCEP_addr);

else send_DT(rcv.SA, —);

66m timeout ∧ (expected_DLPDUs≠none) ∧ (retry_count < max_retry_count)

∧ required_interactions≠rsp_ind

: **if** expected_DLPDUs=RC

then send_RC(remote_DLCEP_addr, DLCEP_addr, !reply_requested, V_C(M), V_C(L)+1);

else send_RC(remote_DLCEP_addr, DLCEP_addr, reply_requested, V_C(M), V_C(L)+1);

***n*0a** DL-DISCONNECT.req

(2≤***n***≤6) **:** **if** class=PEER

then send_DC(remote_DLCEP_addr, DLCEP_addr, reply_requested);

elseif class=PUBLISHER

then send_DC(—, DLCEP_addr, !reply_requested);

start_timer();

***n*0b** internal disconnect condition when a DC DLPDU is appropriate

(2≤***n***≤6) **:** **if** class=PEER

then send_DC(remote_DLCEP_addr, DLCEP_addr, reply_requested);

elseif class=PUBLISHER

then send_DC(—, DLCEP_addr, !reply_requested);

DL-DISCONNECT.ind;

start_timer();

***n*0c** internal disconnect condition when a DC DLPDU is not appropriate

(2≤***n***≤3) **:** DL-DISCONNECT.ind;

start_timer();

***n*0d** rcv.FC=DC ∧ rcv.DA=DLCEP_addr ∧ rcv.SA=remote_DLCEP_addr

(2≤***n***≤6) **:** **if** rcv.DC_p.reply_requested ∧ class=PEER

then send_DC(remote_DLCEP_addr, DLCEP_addr, !reply_requested);

DL-DISCONNECT.ind;

start_timer();

***nn*a** rcv.FC ∈ {EC,RC,CA,CD,ED,DT} ∧ rcv.DA_exists ∧ rcv.DA=DLCEP_addr

(3≤***n***≤6) ∧ rcv.SA_exists ∧ rcv.SA≠remote_DLCEP_addr ∧ class=PEER

(or ***n***=0) **:** send_DC(rcv.SA, DLCEP_addr, !reply_requested);

F.3 FSMs for scheduling

F.4 FSMs for bridges

F.4.1 Overview

F.4.2 The port-state FSM

See C.6 and clause 6 of ISO/IEC 10038.

F.4.3 The bridge-state FSM

See C.6 and clause 6 of ISO/IEC 10038.

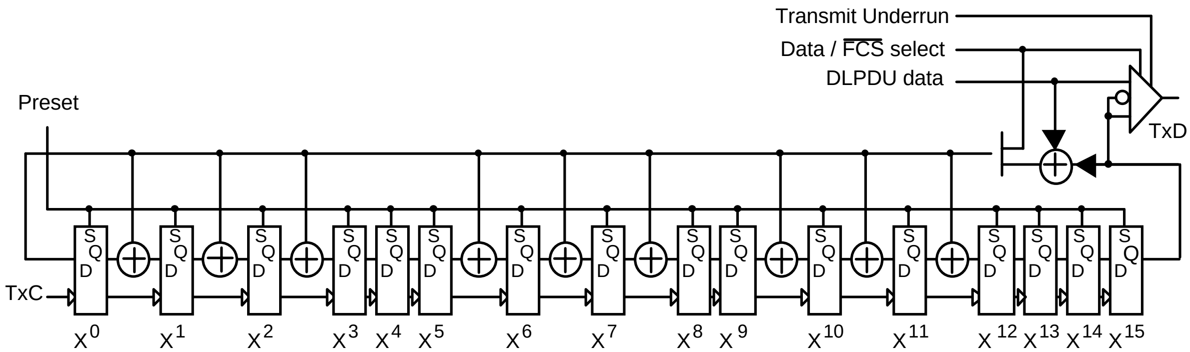
F.4.4 The DLPDU forwarding FSM

F.4.5 The time distribution FSM

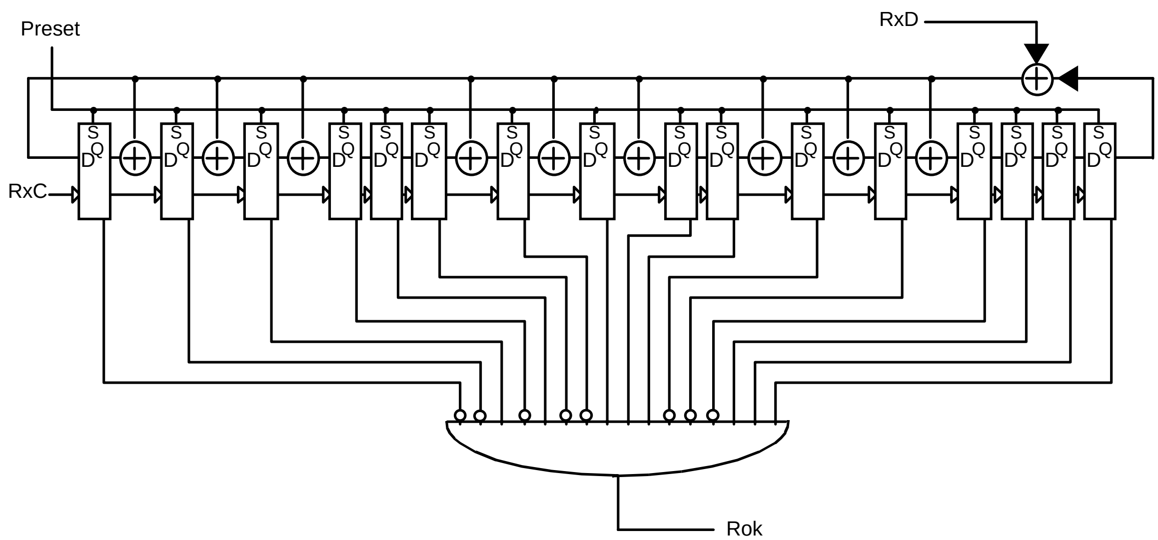
Annex G — (informative) Exemplary FCS implementation

(This annex does not form part of this standard.)

This clause provides an example implementation of FCS generation and FCS syndrome checking.



In this example, the FCS is computed in a register consisting of 16 presettable master-slave flip-flops which are interconnected as a linear feedback shift register, with its least significant bit depicted on the left. The initial preset of the register before transmission serves to include the initial L(X) term in the FCS computation. Feedback is disabled during transmission of the FCS itself, and the FCS is transmitted complemented to provide the final L(X) inclusion in the FCS computation. Also shown is optional logic to inhibit the final complementation and transmit a massively incorrect FCS in the case of a transmitter underrun.



In this example, the residual FCS is computed in a similar register. The Q outputs of the 16 flip-flops are compared to the expected residual value by the 16-input “and” gate, half of whose inputs are complemented.

Annex H — (informative) DLPDU and DL-addressing short-form summaries

(This annex does not form part of this standard.)

This clause provides a number of compact summaries of the various types of DLPDUs and DL-addresses.

H.1 Fields used in short-form summaries

H.1.1 Subfields of the Frame Control (FC) octet

PP	DLPDU and transaction priority (01=URGENT, 10=NORMAL, 11=TIME-AVAILABLE)
L	length of address fields in DLPDUs which have both a long and a short form (0=SHORT — two-octet node.selector with an implicit link=0000_{16}, 1=LONG — four-octet link.node.selector)
F	final transaction in sequence, explicitly returning the token to LAS at the end of the current transaction, : (0=NOT FINAL, 1=FINAL) on sequence-initiating (ES) and token-passing (PT) DLPDUs a third meaning applies: (0= RESTART, 1=CONTINUE) sequence or repetitively-scheduled activities

H.1.2 Nomenclature for address and other fields of DLPDUs

—	symbol indicating an omitted field
d-	prefix indicating a destination address
s-	prefix indicating a source address
o-	prefix indicating an optional field (which is always the last non-omitted field in the DLPDU)
[PP]	brackets indicating implied priority field
DLSAP	a DL-service-access-point address, whose explicit length is specified by the FC's L subfield
DL(SAP)	either a group DL-address or a DLSAP-address, whose explicit length is specified by the FC's L subfield
DLCEP	a DL-address used as a DL-connection-end-point address, whose explicit length is specified by the FC's L subfield
DLCEP/SAP	either a DLCEP-address or a DLSAP-address, whose explicit length is specified by the FC's L subfield
DLCEP/(SAP)	either a group DL-address or a DLSAP-address or a DLCEP-address, whose explicit length is specified by the FC's L subfield
DLSEP	a DL-address used as a DL-scheduling-end-point address, whose explicit length is specified by the FC's L subfield
[d-DLCEP/SAP]	brackets indicating implied d-DLCEP/SAP-address field, whose implied value is the same as that of the explicit or implied s-DLCEP/SAP-address field in the immediately-prior DLPDU, which must have been a CA , CD or ED DLPDU.

[s-DLCEP/SAP]brackets indicating implied s-DLCEP/SAP-address field, whose implied value is the same as that of the explicit d-DLCEP/SAP-address field in the immediately-prior DLPDU, which must have been a CA, CD or ED DLPDU.

H.1.3 Basic parameter and data fields of DLPDUs

link — individual link address (two octets), one component of LONG addresses; also used in time distribution (TD) DLPDUs

node — individual node address (one octet), one-component of SHORT and LONG addresses. Also used as a single octet address in some link-local DLPDUs used for inter-DLE coordination, where it is known as a node DL-address

DLSDU — complete Data Link Service Data Unit (non-null)

pDLSDU — partial or complete DLSDU (non-null)

SPDU — Support Protocol Data Unit (non-null) - a PDU used to convey scheduling and related DLS-support information between DLEs on the local or extended link.

SD-p — multi-octet SD (status and data-description) parameters of the CA, CD, ED and DT DLPDUs (see H.6)

SR-p — parameters of the SR DLPDU (see H.6.4 and H.6.5)

EC-p — multi-octet parameters of the EC DLPDU (see H.7)

DC-p — multi-octet parameters of the DC DLPDU (see H.8.1)

RC-p — multi-octet parameters of the RC DLPDU (see H.8.2)

TD-p — multi-octet parameters of the TD DLPDU (see H.9.1)

RQ-p — multi-octet parameters of the RQ DLPDU (see H.9.2)

RR-p — multi-octet parameters of the RR DLPDU (see H.9.3)

PN-p — multi-octet parameters of the PN DLPDU (see H.10.1)

DD-p — multi-octet parameters of the PT, ES and RI DLPDUs (see H.10.2)

H.2 DLPDU short-form summary grouped by function

H.2.1 Connection-oriented DLPDUs

Establish Connection	1110 LF00	EC		s-DLCEP	s-DLSAP	EC-p	o-DLSDU
	1111 LF00	EC	d-DLCEP/(SAP)	s-DLCEP/SAP	s-DLSAP	EC-p	o-DLSDU
Disconnect Connection	0110 LF00	DC		s-DLCEP	DC-p	o-DLSDU	
	0111 LF00	DC	d-DLCEP/(SAP)	s- DLCEP	DC-p	o-DLSDU	
Reset Connection	0110 LFPP	RC		s-DLCEP	RC-p	o-DLSDU	
	0111 LFPP	RC	d-DLCEP	s-DLCEP	RC-p	o-DLSDU	
Compel Acknowledgment	1010 LFPP	CA	d-DLCEP	—	SD-p	o-pDLSDU	
	1110 LFPP	CA	d-DLCEP	s-DLCEP/SAP	SD-p	o-pDLSDU	
Compel Data	1011 LFPP	CD	d-DLCEP	—	o-SD-p	—	
	1111 LFPP	CD	d-DLCEP	s-DLCEP/SAP	o-SD-p	—	
Exchange Data	1000 LFPP	ED	d-DLCEP	—	SD-p	pDLSDU	
	1100 LFPP	ED	d-DLCEP	s-DLCEP/SAP	SD-p	pDLSDU	
Data	0101 0F00	DT [PP]		[s-DLCEP]	SD-p	o-pDLSDU	
	0101 LFPP	DT		s-DLCEP	SD-p	o-pDLSDU	
	1001 0F00	DT [PP]	[d-DLCEP/SAP]	—	SD-p	o- DLSDU	
	1001 LFPP	DT	d-DLCEP	—	SD-p	o-pDLSDU	
	1001 LFPP	DT	d-DLSAP	—	SD-p	o- DLSDU	
	1101 LFPP	DT	d-DLCEP	s-DLCEP/SAP	SD-p	o-pDLSDU	
Status Response	0001 0F11	SR	—	s-node	o-SR-p	—	

H.2.2 Connectionless DLPDUs

Compel Acknowledgment	1110 LFPP	CA	d-DLSAP	s-DLSAP	SD-p	DLSDU
Compel Data	1111 LFPP	CD	d-DLSAP	s-DLSAP	SD-p	—
Exchange Data	1100 LFPP	ED	d-DLSAP	s-DLSAP	SD-p	DLSDU
Data	1001 0F00	DT [PP]	[d-DLSAP]	—	SD-p	o-DLSDU
	1001 LFPP	DT	d-DLSAP	—	SD-p	o-DLSDU
	1101 LFPP	DT	d-DL(SAP)	s-DLSAP	SD-p	o-DLSDU
Status Response	0001 0F11	SR	—	s-node	o-SR-p	—

NOTE — The same FC values are used for connection-oriented and connectionless DLPDUs. In all cases, the structure of SD-parameters depends on characteristics of the DL-object addressed by the DLPDU’s first explicit or implicit DL-address.

H.2.3 Time Support DLPDUs

Compel Time	0001 0F00	CT	—	—	—	—
Time Distribution	0001 0F01	TD	—	s-node	TD-p	—
Round-trip-delay Query	1100 0F00	RQ	d-node.0	s-node.0	RQ-p	—
Round-trip-delay Reply	1101 0F00	RR	d-node.0	s-node.0	RR-p	—

H.2.4 Link Management DLPDUs

Probe Node-address	0010 0110	PN	d-node	—	PN-p	—
Probe Response	0010 0111	PR	—	[s-node]	—	SPDU
Pass Token	0011 0FPP	PT	d-node	—	DD-p	—
Execute Sequence	1000 LF00	ES	d-DLSEP	—	DD-p	—
Return Token	0011 0100	RT	—	—	—	—
Request Interval	0010 0000	RI	—	—	DD-p	—
Claim LAS	0000 0001	CL	—	s-node	—	—
Transfer LAS	0000 0110	TL	d-node	—	TL-p	—
Status Response	0001 0F11	SR	—	s-node	o-SR-p	—
Wakeup	0000 0000	WK	d-node	—	—	—
Idle	0001 0F10	IDLE	—	—	—	o-DLSDU

H.2.5 Spare and Unusable DLPDU code points

spare	0000 0x11
	0010 0x01
	00xx 1xxx
	0100 xxxx
	101x xx00

unusable	0000 0010
	0000 010x
	0010 001x
	0010 0100
	0011 0000
	0101 1x00
	1001 1x00
	110x 1x00

H.3 DLPDU short-form summary in alphabetic order of DLPDU names

Claim LAS	0000 0001	CL	—	s-node		—	—
Compel Acknowledgment	1010 LFPP	CA	d-DLCEP	—		SD-p	o-pDLSDU
	1110 LFPP	CA	d-DLCEP	s-DLCEP/SAP		SD-p	o-pDLSDU
	1110 LFPP	CA	d-DLSAP	s-DLSAP		SD-p	DLSDU
Compel Data	1011 LFPP	CD	d-DLCEP	—		o-SD-p	—
	1111 LFPP	CD	d-DLCEP	s-DLCEP/SAP		o-SD-p	—
	1111 LFPP	CD	d-DLSAP	s-DLSAP		SD-p	—
Compel Time	0001 0F00	CT	—	—		—	—
Disconnect Connection	0110 LF00	DC		s-DLCEP		DC-p	o-DLSDU
	0111 LF00	DC	d-DLCEP/(SAP)	s- DLCEP		DC-p	o-DLSDU
Data	0101 0F00	DT [PP]		[s-DLCEP]		SD-p	o-pDLSDU
	0101 LFPP	DT		s-DLCEP		SD-p	o-pDLSDU
	1001 0F00	DT [PP]	[d-DLCEP/SAP]	—		SD-p	o- DLSDU
	1001 LFPP	DT	d-DLCEP	—		SD-p	o-pDLSDU
	1001 LFPP	DT	d-DLSAP	—		SD-p	o- DLSDU
	1101 LFPP	DT	d-DLCEP	s-DLCEP/SAP		SD-p	o-pDLSDU
	1101 LFPP	DT	d-DL(SAP)	s-DLSAP		SD-p	o- DLSDU
Establish Connection	1110 LF00	EC		s-DLCEP	s-DLSAP	EC-p	o-DLSDU
	1111 LF00	EC	d-DLCEP/(SAP)	s-DLCEP/SAP	s-DLSAP	EC-p	o-DLSDU
Exchange Data	1000 LFPP	ED	d-DLCEP	—		SD-p	pDLSDU
	1100 LFPP	ED	d-DLCEP	s-DLCEP/SAP		SD-p	pDLSDU
	1100 LFPP	ED	d-DLSAP	s-DLSAP		SD-p	DLSDU
Execute Sequence	1000 LF00	ES	d-DLSEP	—		DD-p	—
Idle	0001 0F10	IDLE	—	—		—	o-DLSDU
Pass Token	0011 0FPP	PT	d-node —			DD-p	—
Probe Node-address	0010 0110	PN	d-node —			PN-p	—
Probe Response	0010 0111	PR	—	[s-node]		—	o-SPDU
Request Interval	0010 0000	RI	—	—		DD-p	—
Reset Connection	0110 LFPP	RC		s-DLCEP		RC-p	o-DLSDU
	0111 LFPP	RC	d-DLCEP	s-DLCEP		RC-p	o-DLSDU
Return Token	0011 0100	RT	—	—		—	—
Round-trip-delay Query	1100 0F00	RQ	d-node.0	s-node.0		RQ-p	—
Round-trip-delay Reply	1101 0F00	RR	d-node.0	s-node.0		RR-p	—
Status Response	0001 0F11	SR	—	s-node		o-SR-p	—
Time Distribution	0001 0F01	TD	—	s-node		TD-p	—
Transfer LAS	0000 0110	TL	d-node	—		TL-p	—
Wakeup	0000 0000	WK	d-node	—		—	—

NOTE — The same FC values are used for connection-oriented and connectionless DLPDUs. In all cases, the structure of SD-parameters depends on characteristics of the DL-object addressed by the DLPDU's first explicit or implicit DL-address.

H.4 DLPDU short-form summary in alphabetic order of DLPDU acronyms

Compel Acknowledgment	1010 LFPP	CA	d-DLCEP	—		SD-p	o-pDLSDU
	1110 LFPP	CA	d-DLCEP	s-DLCEP/SAP		SD-p	o-pDLSDU
	1110 LFPP	CA	d-DLSAP	s-DLSAP		SD-p	DLSDU
Compel Data	1011 LFPP	CD	d-DLCEP	—		o-SD-p	—
	1111 LFPP	CD	d-DLCEP	s-DLCEP/SAP		o-SD-p	—
	1111 LFPP	CD	d-DLSAP	s-DLSAP		SD-p	—
Claim LAS	0000 0001	CL	—	s-node		—	—
Compel Time	0001 0F00	CT	—	—		—	—
Disconnect Connection	0110 LF00	DC		s-DLCEP		DC-p	o-DLSDU
	0111 LF00	DC	d-DLCEP/(SAP)	s- DLCEP		DC-p	o-DLSDU
Data	0101 0F00	DT [PP]		[s-DLCEP]		SD-p	o-pDLSDU
	0101 LFPP	DT		s-DLCEP		SD-p	o-pDLSDU
	1001 0F00	DT [PP]	[d-DLCEP/SAP]	—		SD-p	o- DLSDU
	1001 LFPP	DT	d-DLCEP	—		SD-p	o-pDLSDU
	1001 LFPP	DT	d-DLSAP	—		SD-p	o- DLSDU
	1101 LFPP	DT	d-DLCEP	s-DLCEP/SAP		SD-p	o-pDLSDU
	1101 LFPP	DT	d-DL(SAP)	s-DLSAP		SD-p	o- DLSDU
Establish Connection	1110 LF00	EC		s-DLCEP	s-DLSAP	EC-p	o-DLSDU
	1111 LF00	EC	d-DLCEP/(SAP)	s-DLCEP/SAP	s-DLSAP	EC-p	o-DLSDU
Exchange Data	1000 LFPP	ED	d-DLCEP	—		SD-p	pDLSDU
	1100 LFPP	ED	d-DLCEP	s-DLCEP/SAP		SD-p	pDLSDU
	1100 LFPP	ED	d-DLSAP	s-DLSAP		SD-p	DLSDU
Execute Sequence	1000 LF00	ES	d-DLSEP	—		DD-p	—
Idle	0001 0F10	IDLE	—	—		—	o-DLSDU
Probe Node-address	0010 0110	PN	d-node	—		PN-p	—
Probe Response	0010 0111	PR	—	[s-node]		—	o-SPDU
Pass Token	0011 0FPP	PT	d-node —			DD-p	—
Reset Connection	0110 LFPP	RC		s-DLCEP		RC-p	o-DLSDU
	0111 LFPP	RC	d-DLCEP	s-DLCEP		RC-p	o-DLSDU
Request Interval	0010 0000	RI	—	—		DD-p	—
Round-trip-delay Query	1100 0F00	RQ	d-node.0	s-node.0		RQ-p	—
Round-trip-delay Reply	1101 0F00	RR	d-node.0	s-node.0		RR-p	—
Return Token	0011 0100	RT	—	—		—	—
Status Response	0001 0F11	SR	—	s-node		o-SR-p	—
Time Distribution	0001 0F01	TD	—	s-node		TD-p	—
Transfer LAS	0000 0110	TL	d-node	—		TL-p	—
Wakeup	0000 0000	WK	d-node	—		—	—

NOTE — The same FC values are used for connection-oriented and connectionless DLPDUs. In all cases, the structure of SD-parameters depends on characteristics of the DL-object addressed by the DLPDU’s first explicit or implicit DL-address.

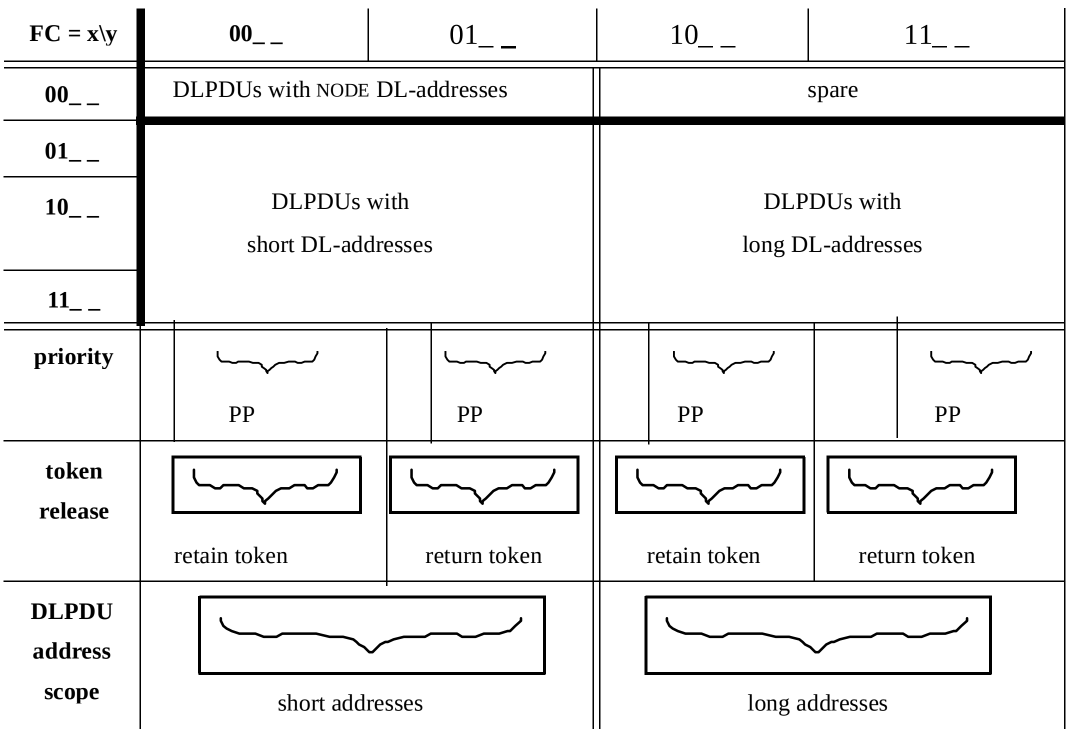
H.5 DLPDU FC code-point assignment matrix — overview and detail

H.5.1 DLPDU FC code-point assignment overview

The first octet of each DLPDU is a frame control (FC) octet which specifies the format of the DLPDU, including the length of all address fields in the DLPDU, and the number and specific roles of the address fields explicitly present in the DLPDU. The FC octet also specifies, where appropriate, the DLPDU’s priority.

The encoding of the FC octet is largely orthogonal, as is shown in Figure H.1. The exact function of each FC code point is shown in Tables H.1 and H.2. In this figure and these tables, the high-order four bits of the FC octet are shown in the row heading on the left side of the figure or table, and the low-order four bits are shown in the column heading at the top of the figure or table.

The blocks shown in Figure H.1 indicate the major functions of that section of the FC code point assignment matrix; a small number of unrelated FC code points are also included in each block.



In Figure H.1, the PP bits take only the values 01, 10 and 11. The PP bits specify the priority of the DLPDU.

Table H.1 shows the generic assignment of the FC code points to the different DLPDUs. It also shows

a) which address fields are explicitly present in the DLPDU (d for destination, s for source, and dss for the three addresses of a connection establishment DLPDU), and which are implicitly present (– for implicit source address); and

b) the common length of each of those d and s address fields, either long (four-octet DL-addresses), or short (one-octet NODE DL-addresses in the upper one-quarter of the table; two-octet DL-addresses in the lower three-quarters of the table, all implicitly on the local link).

Table H.1 — Generic Assignment of FC Code Points

FC=x\y	yyyy	00_ _				01_ _				10_ _				11_ _			
xxxx		**00**	**01**	**10**	**11**	**00**	**01**	**10**	**11**	**00**	**01**	**10**	**11**	**00**	**01**	**10**	**11**
00_ _	**00**	WK d-	CL -s	don’t use		don’t use	don’t use	TL d-									
	01	CT	TD -s	Idle	SR [d]s	CT	TD -s	Idle	SR [d]s								
	10	RI		don’t use	don’t use	don’t use		PN d-	PR -[s]								
	11	don’t use	PT d-			RT	PT d-										
01_ _	**00**																
	01	DT [s]	DT s			DT [s]	DT s			don’t use	DT s			don’t use	DT s		
	10	DC s	RC s			DC s	RC s			DC s	RC s			DC s	RC s		
	11	DC ds	RC ds			DC ds	RC ds			DC ds	RC ds			DC ds	RC ds		
10_ _	**00**	ES d-	ED d-			ES d-	ED d-			ES d-	ED d-			ES d-	ED d-		
	01	DT [d]-	DT d-			DT [d]-	DT d-			don’t use	DT d-			don’t use	DT d-		
	10		CA d-				CA d-				CA d-				CA d-		
	11		CD d-				CD d-				CD d-				CD d-		
11_ _	**00**	RQ ds	ED ds			RQ ds	ED ds			don’t use	ED ds			don’t use	ED ds		
	01	RR ds	DT ds			RR ds	DT ds			don’t use	DT ds			don’t use	DT ds		
	10	EC ss	CA ds			EC ss	CA ds			EC ss	CA ds			EC ss	CA ds		
	11	EC dss	CD ds			EC dss	CD ds			EC dss	CD ds			EC dss	CD ds		

H.5.2 DLPDU code-point assignment rationale

The structure of the frame-control code points is quite regular.

For example

a) FC_3 is the L subfield, specifying the length of all addresses in the DLPDU:

when $FC_3 = 0$ all addresses are SHORT (2 octets when $FC_{7654} \geq 0100$; 1 octet otherwise);

when $FC_3 = 1$ all addresses are LONG (4 octets when $FC_{7654} \geq 0100$).

b) FC_2 is the F subfield, specifying the location of the token at the end of the current transaction: In general, when $FC_2 = 0$ the token stays with the current token holder;

when $FC_2 = 1$ the token returns to the LAS, or transfers to a new LAS, at the end of the current transaction.

c) When $FC_{7654} \geq 0011$ and $FC_{10} \neq 00$, then FC_{10} is the PP subfield.

Table H.2 shows the individual assignment of the FC code points to the different DLPDUs.

Table H.2 — Individual Assignment of FC Code Points

FC=x\y	**yyyy**	**00_ _**				**01_ _**				**10_ _**				**11_ _**			
xxxx		**00**	**01**	**10**	**11**	**00**	**01**	**10**	**11**	**00**	**01**	**10**	**11**	**00**	**01**	**10**	**11**
00_ _	**00**	WK	CL	don't use	spare	don't use	don't use	TL	spare	spare	spare	spare	spare	spare	spare	spare	spare
	01	CT	TD	Idle	SR	CT	TD	Idle	SR	spare	spare	spare	spare	spare	spare	spare	spare
	10	RI	spare	don't use	don't use	don't use	spare	PN	PR	spare	spare	spare	spare	spare	spare	spare	spare
	11	don't use	PT	PT	PT	RT	PT	PT	PT	spare	spare	spare	spare	spare	spare	spare	spare
01_ _	**00**	spare	spare	spare	spare	spare	spare	spare	spare	spare	spare	spare	spare	spare	spare	spare	spare
	01	DT	DT	DT	DT	DT	DT	DT	DT	don't use	DT	DT	DT	don't use	DT	DT	DT
	10	DC	RC	RC	RC	DC	RC	RC	RC	DC	RC	RC	RC	DC	RC	RC	RC
	11	DC	RC	RC	RC	DC	RC	RC	RC	DC	RC	RC	RC	DC	RC	RC	RC
10_ _	**00**	ES	ED	ED	ED	ES	ED	ED	ED	ES	ED	ED	ED	ES	ED	ED	ED
	01	DT	DT	DT	DT	DT	DT	DT	DT	don't use	DT	DT	DT	don't use	DT	DT	DT
	10	spare	CA	CA	CA	spare	CA	CA	CA	spare	CA	CA	CA	spare	CA	CA	CA
	11	spare	CD	CD	CD	spare	CD	CD	CD	spare	CD	CD	CD	spare	CD	CD	CD
11_ _	**00**	RQ	ED	ED	ED	RQ	ED	ED	ED	don't use	ED	ED	ED	don't use	ED	ED	ED
	01	RR	DT	DT	DT	RR	DT	DT	DT	don't use	DT	DT	DT	don't use	DT	DT	DT
	10	EC	CA	CA	CA	EC	CA	CA	CA	EC	CA	CA	CA	EC	CA	CA	CA
	11	EC	CD	CD	CD	EC	CD	CD	CD	EC	CD	CD	CD	EC	CD	CD	CD

H.6 SD-parameters (status and data-description parameters) of CA, CD, ED and DT DLP-DUs

H.6.1 SD-p: Basic DLC parameters for CA, CD, ED and DT DLPDUs

format code	format summary	description
A:	(null)	for use when all DLC directions are UNORDERED or not used, or when no pDLSDU or status is being conveyed
B:	(J, K, T, RR, SSS)	sent by a PEER DLC when sequence numbers are required, segmentation is never needed, and all window sizes are £3
C:	(RRRR, UUUU), (J, K, T, SSSSS), (TTTT, VVVV)	sent by a PEER DLC when one of the two directions is either DISORDERED or CLASSICAL, sequence numbers are required in the other direction, and segmentation may be needed. may also be used in place of format B, D or E
D:	(RRRR, UUUU), (J, K, T, SSSSS)	sent by a PEER DLC when one of the two directions is either DISORDERED or CLASSICAL, sequence numbers are required in the other direction, segmentation is never needed, and at least one window size is >3. may also be used in place of format E
E:	(RRRR, UUUU)	sent by a SUBSCRIBER, or the receiver of an ORDERED PEER DLC, to request a retransmission when segmentation may be needed and when the reverse direction is either UNORDERED or not used
F:	(J=0, K=0, T, SSSSS), (TTTT, VVVV)	sent by a PUBLISHER or PEER DLC when sequence numbers are required, segmentation may be needed, and the reverse direction is either UNORDERED or not used
G:	(J=0, K=0, T, SSSSS)	sent by a PUBLISHER or PEER DLC when sequence numbers are required, segmentation is never needed, and the reverse direction is either UNORDERED or not used

NOTES —

1. Only formats A, C, F and G need be supported. Formats B, D and E are all optimizations of format C, and need not be supported.

2. CA, CD, DT and ED DLPDUs which provide connection-mode services use formats A through G.

Subfield Name	Subfield Description
J	Boolean indicating whether retransmission of segment UUUU of the NDR’th DLSDU is requested.
K	Boolean indicating whether all DLSDUs prior to the NDR’th DLSDU are being acknowledged.
T	Boolean indicating whether the associated DLSDU originated in a buffer with associated timeliness criteria, and that those timeliness criteria were met.
RR	2-bit residue (i.e., the three low-order bits) of the sequence number NDR of a received DLSDU, or of the DLSDU before a missing DLSDU.
RRRR	4-bit residue (i.e., the four low-order bits) of the sequence number NDR of a received DLSDU, or of the DLSDU before a missing DLSDU.
SSS	3-bit residue (i.e., the three low-order bits) of the sequence number NDS of this DLSDU, or of the highest-numbered DLSDU sent on the DLC.
SSSSS	5-bit residue (i.e., the five low-order bits) of the sequence number NDS of this DLSDU, or of the highest-numbered DLSDU sent on the DLC.
UUUU	4-bit zero-based segment number RSN of the DLSDU segment which needs to be retransmitted.
TTTT	4-bit zero-based total number TNS of DLSDU segments in the DLSDU, one segment of which is conveyed in the associated DLPDU.
VVVV	4-bit zero-based segment number ASN of the DLSDU segment conveyed in the associated DLPDU.
Z...Z	one or more bits of zero, as indicated by the number of “Z”s, reserved for future standard use.

H.6.2 SD-p: DLSDU source time stamp for CA, CD, ED and DT DLPDUs

	low-order octets of DL-time, with a unit weight of 2^{-5} ms					
J	null					
K	null				two low-order octets of DL-time	
L	null			three low-order octets of DL-time		
M	DL-time					
	one octet	one octet	one octet	one octet	one octet	one octet

As in all OSI protocols, all multi-octet numeric fields are transmitted most-significant-octet first.

H.6.3 SD-p: Unitdata parameters and Delayed-Reply parameters

format code	format summary	description
P:	(null)	for non-acknowledged unitdata DLPDUs with DL(SAP)s
R:	(ZZ,PP,YYYY)	for acknowledged-unitdata and exchange-unitdata request CA, CD and ED DLPDUs, all with DLSAPs
U:	(XXXX,YYYY)	for unitdata-acknowledgment and exchange-unitdata-reply DT and SR DLPDUs, all with DLSAPs
NOTES — CA, CD and ED DLPDUs which provide connectionless services use only format R; DT DLPDUs with two DL-addresses which provide connectionless services use only format P; DT DLPDUs with one explicit or implicit destination DL-address and an implicit source DL-address which provide connectionless services use only format U; SR DLPDUs use format U.		

Subfield Name Subfield Description

PP 2-bit DLL priority of the DLSDU conveyed in the associated CA, CD or ED DLPDU, as previously specified (01=URGENT, 10=NORMAL, 11=TIME-AVAILABLE).

XXXX 4-bit reply status as specified in table H.3 for SR and table H.4 for appropriate DT DLPDUs.

YYYY 4-bit not-yet-confirmed

ZZ two bits of zero, reserved for future standard use.

H.6.4 SR-p: Status Response parameters

format code	format summary	description
X:	(null)	
Y:	(XXXX, ZZZZ)	for unitdata-acknowledgment and exchange-unitdata-reply error SR DLPDUs, all with DLSAPs

Subfield Name Subfield Description

XXXX 4-bit reply status as specified in table H.3 for SR DLPDUs.

ZZZZ four bits of zero, reserved for future standard use.

H.6.5 Reply status for SR DLPDUs, and for unitdata-acknowledgment and exchange-unitdata-reply DT DLPDUs

Table H.3 — Reply status for SR DLPDUs

Short Name	Definition	hexadecimal coding
RR	failure —resource limitation in responder	4
RI	failure —responder DLE-class incompatible with this request	9
RTR	failure —LAS transfer rejected	B
BF	failure —fault in intermediary bridge	C
BR	temporary failure —resource limitation in intermediary bridge	D
BOK	interim success — intermediary bridge is forwarding transaction	E

Table H.4 — Reply status for unitdata-acknowledgment and exchange-unitdata-reply DT DLPDUs

Short Name	Definition	hexadecimal coding
OK	success — no reply data available at responder or service does not provide this information	0
OK_U	success — URGENT priority reply data available at responder	1
OK_N	success — NORMAL priority reply data available at responder	2
OK_TA	success — TIME-AVAILABLE priority reply data available at responder	3
RR	failure —resource limitation in responder — no reply data available at responder or service does not provide this information (note 1)	4
RR_U	failure —resource limitation in responder — URGENT priority reply data available at responder (note 1)	5
RR_N	failure —resource limitation in responder — NORMAL priority reply data available at responder (note 1)	6
RR_TA	failure —resource limitation in responder — TIME-AVAILABLE priority reply data available at responder	7
RF	failure —fault in responder	8
RI	failure —responder DL(SAP)-role incompatible with this DLPDU	9
RA	failure — response restricted to a different peer DLSAP-address	A
—	reserved for compatibility with the SR DLPDU's reply status — not available for other use	B
BF	failure —fault in intermediary bridge	C
BR	failure —resource limitation in intermediary bridge	D
BOK	reserved for interim success — intermediary bridge is forwarding transaction	E
DR	interim success — delayed reply; end station needs more time to prepare response	F

H.7 EC parameters of EC DLPDUs

EC-p:	(sender's R, NNN, Q, VVV),	-- 1 octet
	(sender's CC, PP, SS, XX),	-- 1 octet
	(sender's MCD_{CR}),	-- 2 octets
	(sender's MCD_D),	-- 2 octets
	(sender's TT_S, A_S, B_S, $WWWW_S$),	-- 1 octet
	(sender's FFF_S, E_S, reserved, G_S, HH_S),	-- 1 octet
	(sender's $M\ldots M_S$),	-- 2 octets
	(sender's TT_R,A_R, B_R, $WWWW_R$)	-- 1 octet
	(sender's FFF_R, E_R, reserved, G_R, HH_R),	-- 1 octet
	(sender's $M\ldots M_R$)	-- 2 octets

As in all OSI protocols, all multi-octet numeric fields are transmitted most-significant-octet first.

R Boolean requiring return of a corresponding DLPDU (1=reply DLPDU requested)

NNN publisher-DLCEP-address reuse-discriminator, or zero for non-publisher DL-addresses

Q DLL path diversity (0=ANY-PATH, 1=THIS-PATH)

VVV DLL protocol version number (always = 1 for this version)

CC DLCEP class (01=PEER, 10=PUBLISHER, 11=SUBSCRIBER)

PP DLL (and transaction) priority, as previously specified (01=URGENT, 10=NORMAL, 11=TIME-AVAILABLE)

SS minimum DLL address size (01=LONG, 10=SHORT, 11=VERY-SHORT)

XX DLPDU authentication (00=ORDINARY, 10=SOURCE, 11=MAXIMAL)

MCD_{CR} negotiated maximum confirm delay for DL-CONNECT and DL-RESET requests
$1 \le D\ldots D \le 60\,000$, or 0xFFFF, where 0xFFFF represents the choice UNLIMITED.

MCD_D negotiated maximum confirm delay for DL-DATA requests
$1 \le D\ldots D \le 60\,000$, or 0xFFFF, where 0xFFFF represents the choice UNLIMITED.

For the following parameters, an S subscript indicates the sender's desired attributes as a source of DLS-user-data on the DLC, and an R subscript indicates the sender's desired attributes as a receiver of DLS-user-data on the DLC.

TT DLCEP features (00=UNORDERED, 01=ORDERED, 10=DISORDERED, 11=CLASSICAL); =00 when M…M=00

A Boolean requiring residual activity in the specified (sender-to-receiver or receiver-to-sender) direction (1=residual activity required)

B Boolean local buffer/queue binding in the specified (sender-to-receiver or receiver-to-sender) direction is to a queue (=0) or to a buffer (=1)

WWWW window size (number of DLPDUs retained for possible retransmission;
= 1 for BUFFER or UNORDERED; = min(K, 15) for QUEUE-k; default = 4 for implicit queue (unless overridden by local DL-management; =0 when M…M=0)

FFF SD-p format (see H.6) in the specified direction (0=format A, 1=format B, 2=format C, 3=reserved, 4=format D, 5=format E, 6=format F, 7=format G)

NOTE – Formats R and U (see H.6) are not used with the connection-oriented DLS.

E Boolean indicating desire/ability to have sent/received ED DLPDUs contain a pDLSDU (1=pDLSDU desired/permitted in ED DLPDUs on this DLC) reserved one or more bits of zero, reserved for future standard use

G Boolean indicating the presence (=1) or absence (=0) of a timeliness parameter in transmitted DLPDUs in the specified direction

HH time-stamp-format in the specified direction (0=format J, 1=format K, 2=format L, 3=format M)

M…M maximum DLSDU size $0 \le M\ldots M \le 16 \times (M\ldots M_{PP})$, based on the DLL priority PP, where $M\ldots M_{Urgent}$=64, $M\ldots M_{Normal}$=128 and $M\ldots M_{Time\text{-}Available}$=256.

H.8 Parameters of DC and RC DLPDUs

H.8.1 DC parameters of DC DLPDUs

DC-p:	(R, reserved, VVV),	-- 1 octet
	(RRRRRRRR)	-- 1 octet

R Boolean requiring return of a corresponding DLPDU (1=return DLPDU requested)

reserved one or more bits of zero, reserved for future standard use

VVV DLL protocol version number (always = 1 for this version)

RRRRRRRR reason for disconnect (see H.8.3)

H.8.2 RC parameters of RC DLPDUs

RC-p:	(R, reserved, VVV),	-- 1 octet
	(RRRRRRRR),	-- 1 octet
	XXXSSSSS),	-- 1 octet

R Boolean requiring return of a corresponding DLPDU (1=return DLPDU requested)

reserved one or more bits of zero, reserved for future standard use

VVV DLL protocol version number (always = 1 for this version)

RRRRRRRR reason for reset (see H.8.3)

XXXSSSSS 5-bit residue of the sequence number N_S preceding the next sequence number to be sent on the DLC, if any, with an arbitrary value for the high-order three bits of the octet.

H.8.3 DC and RC reason coding

code	Reason for disconnect or reset	Reason Class
00	user-originated disconnection — normal condition	user-originated
02	user-originated disconnection — abnormal condition	disconnection
1E	user-originated disconnection or connection rejection — reason unspecified	(00 .. 1F)
20	user-originated connection rejection — connection not authorized, permanent condition	user-originated
21	user-originated connection rejection — unacceptable QoS, permanent condition	connection rejection
22	user-originated connection rejection — non-QoS reason, permanent condition	(20 .. 3F)
24	user-originated connection rejection — transient condition	
40	provider-originated disconnection — incorrect DLCEP pairing, permanent condition	
41	provider-originated disconnection — wrong publisher-DLCEP-address reuse -discriminator, permanent condition	
42	provider-originated disconnection — other permanent condition	provider-originated
43	provider-originated disconnection — wrong DLPDU format or parameters, permanent condition	disconnection (40 .. 5F)
44	provider-originated disconnection — wrong DLSDU size, permanent condition	
45	provider-originated disconnection — transient condition	
46	provider-originated disconnection — timeout	
5E	provider-originated disconnection or connection rejection — reason unspecified	
60	provider-originated connection rejection — DL(SAP) address unknown	
62	provider-originated connection rejection — DLSAP unreachable, permanent condition	provider-originated
64	provider-originated connection rejection — DLSAP unreachable, transient condition	connection rejection
65	provider-originated connection rejection — inconsistent DLCEP state, permanent condition	(60 .. 7D)
66	provider-originated connection rejection — QoS unavailable, permanent condition	
68	provider-originated connection rejection — QoS unavailable, transient condition	
7E	disconnection or connection rejection, unknown origin — reason unspecified	unknown origin disconnect (7E .. 7F)
80	user-originated reset — resynchronization after user timeout	
82	user-originated reset — resynchronization after user-detected user-state inconsistencies	user-originated reset
9E	user-originated reset — reason unspecified	(80 .. BF)
C0	provider-originated reset — resynchronization after activation of a DL-management-established DLCEP	
C2	provider-originated reset — resynchronization after timeout	
C4	provider-originated reset — resynchronization after maximum number of retransmission requests or attempts	provider-originated reset
C6	provider-originated reset — resynchronization after detected sequence number error	(C0 .. FD)
C8	provider-originated reset — resynchronization after other detected DLCEP state inconsistencies	
FC	provider-originated reset — reason unspecified	
FE	reset, unknown origin — reason unspecified	unknown origin reset (FE .. FF)

H.9 Parameters of TD, RQ and RR DLPDUs

C(NT) is the current value of the nodetimer, local to each DLE, zeroed at DLE startup.

NOTE — The three low-order octets of C(NT), or of a value derived from C(NT),

a) are appended to the TD, RQ and RR DLPDUs at the moment of transmission, with an approximately fixed delay relative to transmission of the end-of-frame;

b) are appended to the DLPDU at the moment of reception, with an approximately fixed delay relative to receipt of the end-of-frame.

Thus the TD, RQ and RR DLPDUs can be considered to have three distinct stages in their formation:

i) the DLPDU header as it exists in a transmit queue awaiting the opportunity to transmit, without either of its last two C(NT)-based fields appended;

ii) the DLPDU as it is presented to and reported by the PhL, without its last C(NT)-based field appended; and

iii) the DLPDU as it exists in a receive queue awaiting post-reception processing by upper-DLL software, with all C(NT)-based fields appended.

N(NT) is a previously-recorded value of some DLE’s C(NT).

V(DLTO) (see 5.7.1.21) is the DL-time-offset, which equals the current value of DL-time minus (V(LSTO) + C(NT).

As in all OSI protocols, all multi-octet numeric fields are transmitted most-significant-octet first.

H.9.1 TD-parameters of TD DLPDUs

TD-p:	(L…L),	-- 2 octets,
	(TTT, LLL, SS),	-- 1 octet
	V_S(DLTO),	-- 7 octets,
	(V_S(DLTO) + V_S(LSTO)+C_S(NT)),	-- 7 octets,
	(A…A),	-- 3 octets, appended at moment of transmission
	($C_R(NT)_{23..0}$)	-- 3 octets, appended at moment of reception

V_S(TSL) 16-bit link-id of the root link of the spanning (sub-)tree from which the DL-time sense originated, that is, the link-id of the time-source, V_S(TSL)

L…L 16-bit link-id of the root link of the spanning (sub-)tree from which the DL-time sense originated, that is, the link-id of the time-source, V_S(TSL)

TTT limiting time-synchronism class between time-source and sender, $N_S(TQ)_{7-5}$ (000=NONE, 001=1 s, 010=100 ms, 011=10 ms, 100=1 ms, 101=100 µs, 110=10 µs, 111=1 µs)

LLL number of intervening links on the DL-time propagation path from the DL-time source DLE to the sending DLE, $N_S(TQ)_{4-2}$,

0 indicates that the DL-time originates with the DLE itself,

1 indicates that the DL-time originates with another DLE on the local link,

2 indicates that the DL-time originates with a DLE on a link one bridge removed,

and so forth;

SS time-source-type, expressing the method of synchronization of the DL-time source DLE with Universal Coordinated Time (UTC), the world-wide time standard, $N_S(TQ)_{1-0}$

00 some DLE’s node-time, locally generated and not received from an extra-DLS-provider source,

01 local time (not UTC) received (directly or indirectly) from a human source,

10 UTC received (directly or indirectly) from a human source or an unreliable electronic source, or previously (but no longer) received (directly or indirectly) from a reliable electronic source,

11 UTC continuously received (directly or indirectly) from a reliable electronic source.

V_S(DLTO) the DL-time-offset of the sending DLE during DLPDU formation shall be expressed as a signed seven-octet integer encoded most-significant-octet first, where the least significant bit represents a time granularity of approximately 2^{-13} ms

V_S(DLTO) + V_S(LSTO)+$C_S(NT)_1$ the DL-time of the sending DLE at a moment during DLPDU formation, less than 1,1 s prior to transmission, where the least significant bit represents a time granularity of approximately 2^{-13} ms

A…A a small adjustment to account for systemic offsets in the previous subfield, where the least significant bit represents a time granularity of approximately 2^{-13} ms, computed as $C_S(NT)_2 - C_S(NT)_1$, where $C_S(NT)_2$ represents the moment of end-of-transmission

$C_R(NT)_{23..0}$ low-order three octets of TD receiver's nodetimer at the moment of reception. This subfield does not appear on the medium

Table H.5 — Approximate numeric significance of the bits of seven-octet DL-time

octet of DL-time in transmission order	symbolic contents	approximate weight of the low-order bit of the octet
1	0YYYYYYY	1,09 yr
2	DDDDDDDD	1,55 day
3	HHHHHMMM	8,74 min
4	MMMSSSSS	2,05 s
5	Smmmmmmm	8,00 ms
6	mmm•µµµµµ	31,25 µs
7	µµµµµnnn	122 ns

Table H.6 — Approximate numeric significance of the bits of N(NT), A…A, and three-octet C(NT)

octet of short time in transmission order	symbolic contents	approximate weight of the low-order bit of the octet
1	Smmmmmmm	8,00 ms
2	mmm•µµµµµ	31,25 µs
3	µµµµµnnn	122 ns

H.9.2 RQ-parameters of RQ DLPDUs

RQ-p: ($C_S(NT)_{23..0}$) -- 3 octets, appended at the moment of transmission
($C_R(NT)_{23..0}$) -- 3 octets, appended at the moment of reception

$C_S(NT)_{23..0}$ low-order three octets of RQ sender's nodetimer at the moment of RQ DLPDU formation just prior to (or during) transmission, representing the low-order three octets of the DLS-provider's current node-time in units of 2^{-13} ms

$C_R(NT)_{23..0}$ low-order three octets of RQ receiver's nodetimer at moment of reception. This subfield does not appear on the medium

H.9.3 RR-parameters of RR DLPDUs

RR-p:	(TTT, E, reserved),	-- 1 octet
		($N_R(NT)_{23..0}$), -- 3 octets,
	($N_S(NT)_{23..0}$)	-- 3 octets,
	($C_S(NT)_{23..0}$),	-- 3 octets, appended at moment of transmission
	($C_R(NT)_{23..0}$)	-- 3 octets, appended at moment of reception

TTT — time-synchronism class of the replying DLE, $N_S(TQ)_{7\text{-}5}$, encoded as in H.9.2

E — Boolean indicating whether a DLE maintains a real or estimated C(NT); true unless $N_S(TQ)_{7\text{-}5}$ specifies NONE

reserved — one or more bits of zero, reserved for future standard use

$N_R(NT_{23..0})$ — low-order three octets of RR receiver's (RQ sender's) nodetimer at the moment of transmission of the interrogating RQ DLPDU

$N_S(NT_{23..0})$ — low-order three octets of RR sender's (RQ receiver's) nodetimer at the moment of reception of the interrogating RQ DLPDU, adjusted for any systemic offset in the value of the this and the next subfield

$C_S(NT)_{23..0}$ — low-order three octets of RR sender's nodetimer at the moment of RR DLPDU formation just prior to (or during) transmission, representing the low-order three octets of the DLS-provider's current node-time in units of 2^{-13} ms

$C_R(NT)_{23..0}$ — low-order three octets of RR receiver's nodetimer at the moment of reception. This subfield does not appear on the medium

H.10 Parameters of PN, PT, ES and RI DLPDUs

As in all OSI protocols, all multi-octet numeric fields are transmitted most-significant-octet first.

H.10.1 PN-parameters of PN DLPDUs

PN-p:	(SSSS, reserved, VVV),	-- 1 octet
	(GGGG, PPPP),	-- 1 octet
	(TTTTTTTT TTTTTTTT),	-- 2 octets
	(DDDDDDDD),	-- 1 octet
	(MMMMMMMM)	-- 1 octet

SSSS — PhL maximum inter-channel signal skew

reserved — one or more bits of zero, reserved for future standard use

VVV — DLL protocol version number (always = 001 for this version)

GGGG — PhL post-transmission-gap extension units

PPPP — PhL preamble extension units

T…T — slot-time, V(ST), $1 \le$ T…T $\le 4\,095$ octet-durations

D…D — maximum-response-delay, V(MRD), $0 \le$ D…D ≤ 11 slot-times

M…M — minimum-inter-PDU-delay, V(MID), $0 \le$ M…M $\le \min(120, (\text{D…D} - 1) \times \text{T…T})$

H.10.2 DD-parameters (delegation-duration parameters) of PT, ES and RI DLPDUs

DD-p: (DDDDDDDD DDDDDDDD) -- 2 octets

D…D — requested or delegated duration of token usage, V(RD), measured in octet-durations

H.11 Addressing summary extracted from Annex A figures and tables

The figures and tables in this section are reproduced from Annex A. They provide a convenient summary of the structure of DL-addresses and the preassigned ranges and uses of those DL-addresses.

Link Designator	**Sub-link Selector**
2 octets	2 octets

Figure H.2 [Annex A/1] Basic Structure of a DL-address

Universally-administered MAC-addresses			
I/G	0	globally-administered	vendor-administered
1	1	22 bits	24 bits

implied Link Designator (0000)	**Node Designator**	**Sub-node Selector**
	1 octet	1 octet

Figure H.3 [Annex A/2] — Basic Structure of a Sub-link Selector

Locally-administered MAC-addresses		
I/G	1	user-administered
1	1	46 bits

Figure H.5 [Annex A/4] — Basic Structure of MAC-addresses

Link	**Node**	Selector
2 octets	1 octet	1 octet

or

Link	**Node || Selector**
2 octets	2 octets

or

Link || Node || Selector
4 octets

Figure H.4 [Annex A/3] — DL-Address Alternative Structures

a DL-address as a locally-administered MAC-address			
I/G	1	set by DL-management	extended DL-address
1	1	14 bits	32 bits

Figure H.6 [Annex A/5] — Representation of a DL-address as a MAC-address

Table H.7 [Annex A/1] — link || node || selector addressing

link	N | S	assigned use for specified DL-address range
0000	0000 … FFFF	local link DL-addresses as specified in Table A.2.
0001	0000 … 0007	flat non-local group DL-addresses reserved for DLL use
0001	0008 … 003F	flat non-local group DL-addresses reserved for standard AEs
0001 … 003F	0040 … FFFF	flat non-local group DL-addresses available for vendor, user and dynamic assignment
0040 … 007F	0000 … FFFF	flat non-local globally-administered* group DL-addresses — potentially one per vendor
0080	0000 … 0007	flat non-local individual “DLSAP”-addresses reserved for DLL use
0080	0008 … 003F	flat non-local individual DLSAP-addresses reserved for standard AEs
0080 … 00FF	0040 … FFFF	flat non-local individual DLSAP-addresses available for vendor, user and dynamic assignment
0100 0FFF	0000 FFBF	flat non-local individual DL-addresses available for vendor, user and dynamic assignment as DLCEP-addresses and DLSEP-addresses
0FFF	FFC0 … FFF7	flat non-local individual DL-addresses reserved for standard AE DLCEP-addresses and DLSEP-addresses
0FFF	FFF8 … FFFF	flat non-local individual DL-addresses reserved for DLL use as DLCEP-addresses and DLSEP-addresses
1000 … FEFF	0000 … FFFF	Individual links of DL-addresses as specified in Table A.2.

*These group DL-addresses are based on the Organizationally Unique Identifiers specified in the ISO/IEC 8802 family of Local Area Network standards.

Table H.8 [Annex A/2] — link-local node || selector addressing

N	S	assigned use for specified DL-address range
00	00 … FF	local node DL-addresses as specified in Table A.4.
01	00 … 07	flat link-local group DL-addresses reserved for DLL use
01	08 … 3F	flat link-local group DL-addresses reserved for standard AEs
01 … 03	40 … FF	flat link-local group DL-addresses available for vendor, user and dynamic assignment
04	00 … 07	flat link-local individual “DLSAP”-addresses reserved for DLL use
04	08 … 3F	flat link-local individual DLSAP-addresses reserved for standard AEs
04	40 … FF	flat link-local individual DLSAP-addresses available for vendor, user and dynamic assignment
05	00 BF 0F	flat link-local individual DL-addresses available for vendor, user and dynamic assignment as DLCEP-addresses and DLSEP-addresses
0F	C0 … F7	flat link-local individual DL-addresses reserved for standard AE DLCEP-addresses and DLSEP-addresses
0F	F8 … FF	flat link-local individual DL-addresses reserved for DLL use as DLCEP-addresses and DLSEP-addresses
10 … FF	00 … FF	Individual nodes of DL-addresses as specified in Table A.4.

Table H.9 [Annex A/7] — pre-defined flat non-local DL-addresses

link	N || S	assigned use for specified DL-address
0001	0000	the DL-support functions of “all” (see note 1) DLEs on the extended link
0001	0001	the DL-support functions of “all” (see note 1) LM DLEs on the extended link
0001	0002	the DL-support functions of “all” (see note 1) bridge DLEs on the extended link
0001	0003	the DL-bridge functions of “all” (see note 1) bridge DLEs on the extended link
0001	0008	the SMAEs of “all” (see note 1) unconfigured DLEs on the extended link
0001	0009	the SMAEs of “all” (see note 1) DLEs on the extended link
0001	000A	the SMAEs of “all” (see note 1) LM DLEs on the extended link
0001	000B	the SMAEs of “all” (see note 1) bridge DLEs on the extended link
0001	000C	the SLAEs of “all” (see note 1) LoadServers on the extended link
0001	000D	the SLAEs of “all” (see note 1) Loadable Devices on the extended link
0080	0000	reserved for DLL use for a DL-support “DLSAP”-address (see note 1)
0080	0004	the “DLSAP”-address for the DL-bridge functions of the bridge DLE on the extended link which is the root bridge of the spanning tree
0FFF	FFFE	the DLCEP-address for the DLC from the bridge functions of the bridge DLE on the extended link which is the root bridge of the spanning tree to the bridge functions in all other bridge DLEs on the extended link
0FFF	FFFF	reserved for DLL use for a DL-support DLCEP-address or DLSEP-address (see note 1)
NOTE 1 — DLEs which do not recognize LONG DL-addresses are necessarily excluded from these sets		

Table H.10 [Annex A/6] — pre-defined flat link-local DL-addresses

node ||	selector	assigned use for specified DL-address
01	00	the DL-support functions of all DLEs on the link
01	01	the DL-support functions of all LM DLEs on the link
01	02	the DL-support functions of all bridge DLEs on the link
01	03	the DL-bridge functions of all bridge DLEs on the link
01	08	the SMAEs of all unconfigured DLEs on the link
01	09	the SMAEs of all DLEs on the link
01	0A	the SMAEs of all LM DLEs on the link
01	0B	the SMAEs of all bridge DLEs on the link
01	0C	the SLAEs of all LoadServers on the link
01	0D	the SLAEs of all LoadableDevices on the link
04	00	the “DLSAP”-address for the DL-support functions of the DLE on the link which is serving as LAS
04	04	the “DLSAP”-address for the DL-bridge functions of the bridge DLE on the link which is dominant (closest to the root) in the bridge spanning tree
0F	FE	the DLCEP-address for the DLC from the bridge functions of the bridge DLE on the link which is dominant (closest to the root) in the bridge spanning tree to the bridge functions in all other bridge DLEs on the link
0F	FF	the DLCEP-address for the DLC from the DL-support functions of the DLE on the link which is serving as LAS to the DL-support functions of all of the other LM DLEs on the link

Table H.11 [Annex A/3] — link-local node designators

node	assigned use for specified DL-address range
00	local DLE
01 … 0F	unusable
10 … 13	bridge-class DLEs
14 …	link-master-class DLEs
…	unused
… F7	basic-class DLEs
F8 … FB	non-”visitor” DLEs awaiting proper node designator assignment
FC … FF	“visitor” DLEs

Table H.12 [Annex A/4] — node-local selector addressing

selector	assigned use for specified DL-address range
00 … 01	node-local individual “DLSAP”-addresses reserved for DLL use
02 … 07	node-local individual DLSAP-addresses reserved for standard AEs
08 … 1F	node-local individual DLSAP-addresses available for vendor, user and dynamic assignment
20 … F7	node-local individual DL-addresses available for vendor, user and dynamic assignment as DLCEP-addresses and DLSEP-addresses
F8 … FF	node-local individual DL-addresses reserved for use as standard AE DLCEP-addresses and DLSEP-addresses

Table H.13 [Annex A/7]— pre-defined node-local DL-addresses

selector	assigned use for specified DL-address
00	the “DLSAP”-address for the DL-support functions of the node’s DLE
01	the “DLSAP”-address for the DL-bridge functions of the node’s DLE
02	the DLSAP-address for the node’s SMAE
03	the DLSAP-address for the node’s SLAE

Developing and promulgating technically sound consensus standards, recommended practices, and technical reports is one of ISA's primary goals. To achieve this goal the Standards and Practices Department relies on the technical expertise and efforts of volunteer committee members, chairmen, and reviewers.

ISA is an American National Standards Institute (ANSI) accredited organization. ISA administers United States Technical Advisory Groups (USTAGs) and provides secretariat support for International Electrotechnical Commission (IEC) and International Organization for Standardization (ISO) committees that develop process measurement and control standards. To obtain additional information on the Society's standards program, please write:

ISA
Attn: Standards Department
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, NC 27709

ISBN: 1-55617-642-2