

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 4-21: Data-link layer protocol specification – Type 21 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 4-21: Spécification du protocole de la couche de liaison de données –
Éléments de Type 21**





THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2010 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

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

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

About the IEC

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

About IEC publications

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

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,...).

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Liens utiles:

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

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...).

Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

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

Service Clients - webstore.iec.ch/csc

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



IEC 61158-4-21

Edition 1.0 2010-08

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 4-21: Data-link layer protocol specification – Type 21 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 4-21: Spécification du protocole de la couche de liaison de données –
Éléments de Type 21**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XE

ICS 25.04.40; 35.100.20; 35.110

ISBN 978-2-88912-860-0

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

CONTENTS

FOREWORD.....	6
INTRODUCTION.....	8
1 Scope.....	10
1.1 General.....	10
1.2 Specifications.....	10
1.3 Procedures.....	10
1.4 Applicability.....	10
1.5 Conformance.....	11
2 Normative references	11
3 Terms, definitions, symbols and abbreviations.....	11
3.1 Reference model terms and definitions.....	11
3.2 Service convention terms and definitions.....	13
3.3 Common terms and definitions	14
3.4 Symbols and abbreviations.....	17
4 Overview of the data-link protocol.....	18
4.1 General.....	18
4.2 Overview of medium access control	18
4.3 Service assumed from the physical layer	19
4.4 DLL architecture.....	19
4.5 Data type	21
4.6 Local parameters and variables.....	23
5 General structure and encoding.....	38
5.1 Overview	38
5.2 MAPDU structure and encoding.....	38
5.3 Common MAC frame structure, encoding and elements of procedure	39
5.4 Order of bit transmission	47
5.5 Invalid DLPDU.....	47
6 DLPDU structure and procedure	48
6.1 General.....	48
6.2 Common DLPDU Field	48
6.3 DL-DATA Transfer.....	48
6.4 DL-SPDATA Transfer	51
6.5 Network control messages	53
7 DLE elements of procedure	59
7.1 Overall structure.....	59
7.2 DL-protocol machine (DLPM)	59
7.3 DLL management Protocol	69
8 Constants and error codes	102
8.1 General.....	102
8.2 Constants.....	102
8.3 Data-link layer error codes	104
Bibliography.....	105
Figure 1 – Relationships of DLSAPs, DLSAP-addresses, and group DL-addresses	15
Figure 2 – Interaction of PhS primitives with DLE.....	19

Figure 3 – Data-link layer architecture	20
Figure 4 – Common MAC frame format for Type 21 DLPDU	39
Figure 5 – MAC frame format for other protocols.....	39
Figure 6 – Version and Length field	40
Figure 7 – DST_addr field.....	41
Figure 8 – SRC_addr field.....	42
Figure 9 – Frame Control Field	43
Figure 10 – Extension field	45
Figure 11 – DSAP field	46
Figure 12 – Source service access point field	46
Figure 13 – Length of group mask and extension information.....	47
Figure 14 – Group mask option field	47
Figure 15 – Common DLPDU field	48
Figure 16 – Building a DT DLPDU	49
Figure 17 – DT DLPDU structure	49
Figure 18 – SPDT DLPDU structure	52
Figure 19 – NCM_LA DLPDU structure	54
Figure 20 – DLL structure and elements	59
Figure 21 – State transition diagram of the DLPM.....	63
Figure 22 – State transition diagram of DLM	73
Table 1 – DLL components	20
Table 2 – UNSIGNEDn data type	22
Table 3 – INTEGERn data type	23
Table 4 – DLE configuration parameters	24
Table 5 – Queues to support data transfer	25
Table 6 – Variables to support SAP management	26
Table 7 – Variables to support device information management.....	26
Table 8 – DL-entity identifier	27
Table 9 – Device Flags	27
Table 10 – DLM state.....	27
Table 11 – Device Unique Identification	28
Table 12 – Unique identification of device connected to R-port1	28
Table 13 – Unique identification of device connected to R-port2	28
Table 14 – MAC address.....	28
Table 15 – Port information.....	29
Table 16 – Protocol version	29
Table 17 – Device type	30
Table 18 – Device description.....	30
Table 19 – Hop count.....	30
Table 20 – Variables to support managing network information.....	31
Table 21 – Topology	31
Table 22 – Collision count.....	31

Table 23 – Device count	32
Table 24 – Topology change count	32
Table 25 – Last topology change time	32
Table 26 – RNMP device UID	32
Table 27 – RNMS device UID	33
Table 28 – LNM device UID for R-port1	33
Table 29 – LNM device UID for R-port2	33
Table 30 – Network flags	34
Table 31 – Variables and counter to support managing path information	35
Table 32 – Hop count for R-port1 direction	36
Table 33 – Hop count for R-port2 direction	36
Table 34 – Preferred R-port	36
Table 35 – Destination R-port	36
Table 36 – In net count	37
Table 37 – In net time	37
Table 38 – Out net count	38
Table 39 – Out net time	38
Table 40 – Version and Length	41
Table 41 – Destination DL–entity identifier	41
Table 42 – Source DL–entity identifier	42
Table 43 – Frame control	43
Table 44 – Extension	45
Table 45 – Destination service access point	46
Table 46 – source service access point	46
Table 47 – DT DLPDU parameters	49
Table 48 – Primitives exchanged between DLS-user and DLE to send a DT DLPDU	51
Table 49 – Primitives exchanged between DLS-user and DLEs to receive a DT DLPDU	51
Table 50 – SPDT DLPDU Parameters	52
Table 51 – Primitive exchanged between DLS-User and DLEs to send an SPDT DLPDU	53
Table 52 – Primitives exchanged between DLS-user and DLEs to receive an SPDT DLPDU	53
Table 53 – NCM_LA DLPDU parameters	54
Table 54 – NCM_AT DLPDU parameters	55
Table 55 – NCM_LS DLPDU parameters	56
Table 56 – NCM_RS DLPDU parameters	57
Table 57 – NCM_AR DLPDU parameters	58
Table 58 – Primitives exchanged between DLPM and DLS-user	60
Table 59 – Parameters exchanged between DLPM and DLS-user	61
Table 60 – Primitives exchanged between DLPM and DLM	62
Table 61 – Parameters used with primitives exchanged between DLPM and DLM	63
Table 62 – DLPM state table	64
Table 63 – DLPM functions table	68
Table 64 – Primitives exchanged between DLM and DLS-user	70

Table 65 – Parameters used with primitives exchanged between DLM and DLS-user.....	71
Table 66 – Primitive exchanged between DLM and DMAC	71
Table 67 – Parameters used with primitives exchanged between DLM and DMAC	72
Table 68 – Primitive exchanged between DLM and DPHY.....	72
Table 69 – Parameters used with primitives exchanged between DLM and DPHY.....	72
Table 70 – DLM state table	75
Table 71 – DLM function table	100
Table 72 – DLL constants	103
Table 73 – Type 21 DLL error codes	104

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 4-21: Data-link layer protocol specification –
Type 21 elements**

FOREWORD

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

International Standard IEC 61158-4-21:2010 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This standard cancels and replaces IEC/PAS 62573 published in 2008. This first edition constitutes a technical revision.

This bilingual version published in 2012-01 corresponds to the English version published in 2010-08.

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

FDIS	Report on voting
65C/605/FDIS	65C/619/RVD

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

The French version has not been voted upon.

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

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

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

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

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

INTRODUCTION

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

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

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

This 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.

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

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning Type 21 elements and possibly other types given in subclause 4.1, 4.2 and 7.3 as follows:

KR 0789444	[LS]	A communication packet processing apparatus and method for ring topology ethernet network capable of preventing permanent packet looping
KR 0732510	[LS]	Network system
KR 0870670	[LS]	Method for determining a Ring Manager Node

IEC takes no position concerning the evidence, validity and scope of these patent rights.

The holder of these patent rights has assured the IEC that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of these patent rights is registered with IEC. Information may be obtained from:

[LS]: LS Industrial Systems Co., Ltd.
 LS Tower 1026-6
 Hogye-dong, Dongan-gu,
 Anyang-si, Gyeonggi-do 431-848
 Republic of Korea

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://www.iec.ch/tctools/patent_decl.htm) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases for the most up to date information concerning patents.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 4-21: Data-link layer protocol specification – Type 21 elements

1 Scope

1.1 General

The DLL provides basic time-critical data communications between devices in an automated environment. Type 21 provides priority-based cyclic and acyclic data communication using an internal collision-free, full-duplex dual-port Ethernet switch technology. For wide application in various automation applications, Type 21 does not restrict the cyclic/acyclic scheduling policy in the DLL.

1.2 Specifications

This standard describes:

- a) procedures for the timely transfer of data and control information from one data link user entity to a peer user entity, and among the data link entities forming the distributed data link service provider;
- b) procedures for giving communication opportunities based on standard ISO/IEC 8802-3 MAC, with provisions for nodes to be added or removed during normal operation;
- c) structure of the fieldbus data link protocol data units (DLPDUs) used for the transfer of data and control information by the protocol of this standard, and their representation as physical interface data units.

1.3 Procedures

The procedures are defined in terms of:

- a) the interactions between peer data link entities (DLEs) through the exchange of fieldbus DLPDUs;
- b) the interactions between a data link service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives;
- c) the interactions between a DLS-provider and a physical layer service provider in the same system through the exchange of Ph-service primitives.

1.4 Applicability

These procedures are applicable to instances of communication between systems that support time-critical communications services in the data link layer of the OSI or fieldbus reference models, and that 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-deterministic communications needs.

1.5 Conformance

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

2 Normative references

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

IEC 61158-3-21:2010 ¹, *Industrial Communication Networks – Fieldbus specifications – Part 3-21: Data-link layer service definition – Type 21 elements*

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

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

ISO/IEC 8802-3:2000, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

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

3 Terms, definitions, symbols and abbreviations

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

3.1 Reference model terms and definitions

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

3.1.1 called-DL-address	[ISO/IEC 7498-3]
3.1.2 calling-DL-address	[ISO/IEC 7498-3]
3.1.3 centralized multi-end-point-connection	[ISO/IEC 7498-1]
3.1.4 correspondent (N)-entities	[ISO/IEC 7498-1]
correspondent DL-entities (N=2)	
correspondent Ph-entities (N=1)	
3.1.5 demultiplexing	[ISO/IEC 7498-1]
3.1.6 DL-address	[ISO/IEC 7498-3]
3.1.7 DL-address-mapping	[ISO/IEC 7498-1]

¹ To be published.

3.1.8 DL-connection	[ISO/IEC 7498-1]
3.1.9 DL-connection-end-point	[ISO/IEC 7498-1]
3.1.10 DL-connection-end-point-identifier	[ISO/IEC 7498-1]
3.1.11 DL-connection-mode transmission	[ISO/IEC 7498-1]
3.1.12 DL-connectionless-mode transmission	[ISO/IEC 7498-1]
3.1.13 DL-data-sink	[ISO/IEC 7498-1]
3.1.14 DL-data-source	[ISO/IEC 7498-1]
3.1.15 DL-duplex-transmission	[ISO/IEC 7498-1]
3.1.16 DL-facility	[ISO/IEC 7498-1]
3.1.17 DL-local-view	[ISO/IEC 7498-3]
3.1.18 DL-name	[ISO/IEC 7498-3]
3.1.19 DL-protocol	[ISO/IEC 7498-1]
3.1.20 DL-protocol-connection-identifier	[ISO/IEC 7498-1]
3.1.21 DL-protocol-control-information	[ISO/IEC 7498-1]
3.1.22 DL-protocol-data-unit	[ISO/IEC 7498-1]
3.1.23 DL-protocol-version-identifier	[ISO/IEC 7498-1]
3.1.24 DL-relay	[ISO/IEC 7498-1]
3.1.25 DL-service-connection-identifier	[ISO/IEC 7498-1]
3.1.26 DL-service-data-unit	[ISO/IEC 7498-1]
3.1.27 DL-simplex-transmission	[ISO/IEC 7498-1]
3.1.28 DL-subsystem	[ISO/IEC 7498-1]
3.1.29 DL-user-data	[ISO/IEC 7498-1]
3.1.30 flow control	[ISO/IEC 7498-1]
3.1.31 layer-management	[ISO/IEC 7498-1]
3.1.32 multiplexing	[ISO/IEC 7498-3]
3.1.33 naming-(addressing)-authority	[ISO/IEC 7498-3]
3.1.34 naming-(addressing)-domain	[ISO/IEC 7498-3]
3.1.35 naming-(addressing)-subdomain	[ISO/IEC 7498-3]
3.1.36 (N)-entity	[ISO/IEC 7498-1]
DL-entity	
Ph-entity	
3.1.37 (N)-interface-data-unit	[ISO/IEC 7498-1]
DL-service-data-unit (N=2)	
Ph-interface-data-unit (N=1)	

3.1.38 (N)-layer	[ISO/IEC 7498-1]
DL-layer (N=2)	
Ph-layer (N=1)	
3.1.39 (N)-service	[ISO/IEC 7498-1]
DL-service (N=2)	
Ph-service (N=1)	
3.1.40 (N)-service-access-point	[ISO/IEC 7498-1]
DL-service-access-point (N=2)	
Ph-service-access-point (N=1)	
3.1.41 (N)-service-access-point-address	[ISO/IEC 7498-1]
DL-service-access-point-address (N=2)	
Ph-service-access-point-address (N=1)	
3.1.42 peer-entities	[ISO/IEC 7498-1]
3.1.43 Ph-interface-control-information	[ISO/IEC 7498-1]
3.1.44 Ph-interface-data	[ISO/IEC 7498-1]
3.1.45 primitive name	[ISO/IEC 7498-3]
3.1.46 reassembling	[ISO/IEC 7498-1]
3.1.47 recombining	[ISO/IEC 7498-1]
3.1.48 reset	[ISO/IEC 7498-1]
3.1.49 responding-DL-address	[ISO/IEC 7498-3]
3.1.50 routing	[ISO/IEC 7498-1]
3.1.51 segmenting	[ISO/IEC 7498-1]
3.1.52 sequencing	[ISO/IEC 7498-1]
3.1.53 splitting	[ISO/IEC 7498-1]
3.1.54 synonymous name	[ISO/IEC 7498-3]
3.1.55 systems-management	[ISO/IEC 7498-1]

3.2 Service convention terms and definitions

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

- 3.2.1 acceptor**
- 3.2.2 asymmetrical service**
- 3.2.3 confirm (primitive);
requestor.deliver (primitive)**
- 3.2.4 deliver (primitive)**
- 3.2.5 DL-confirmed-facility**
- 3.2.6 DL-facility**
- 3.2.7 DL-local-view**
- 3.2.8 DL-mandatory-facility**
- 3.2.9 DL-non-confirmed-facility**
- 3.2.10 DL-protocol-machine**
- 3.2.11 DL-provider-initiated-facility**
- 3.2.12 DL-provider-optional-facility**
- 3.2.13 DL-service-primitive;
primitive**
- 3.2.14 DL-service-provider**
- 3.2.15 DL-service-user**
- 3.2.16 DLS-user-optional-facility**
- 3.2.17 indication (primitive);
acceptor.deliver (primitive)**
- 3.2.18 multi-peer**
- 3.2.19 request (primitive);
requestor.submit (primitive)**
- 3.2.20 requestor**
- 3.2.21 response (primitive);
acceptor.submit (primitive)**
- 3.2.22 submit (primitive)**
- 3.2.23 symmetrical service**

3.3 Common terms and definitions

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

3.3.1

DL-segment, link, local link

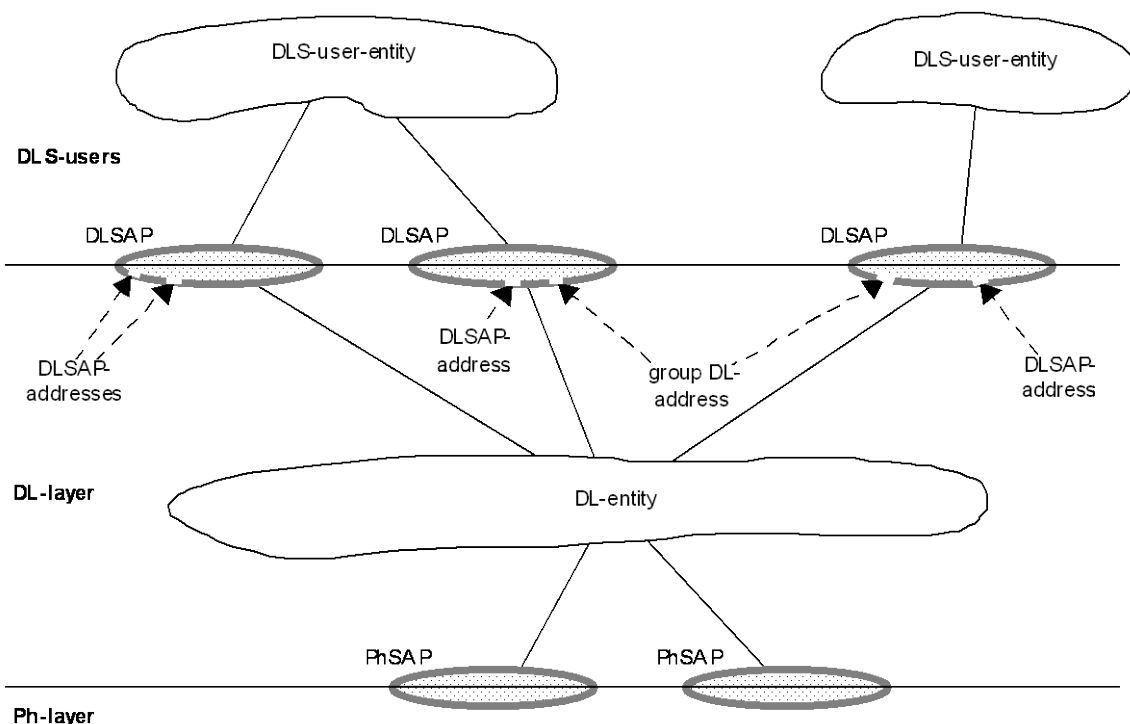
single data link (DL) subnetwork in which any of the connected data link entities (DLEs) may communicate directly, without any intervening data link relaying, whenever all of those DLEs that are participating in an instance of communication are simultaneously attentive to the DL-subnetwork during the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

3.3.2

data-link service access point (DLSAP)

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

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



NOTE 1 DLSAPs and physical layer service access points (PhSAPs) are depicted as ovals spanning the boundary between two adjacent layers.

NOTE 2 DL-addresses are depicted as designating small gaps (points of access) in the DLL portion of a DLSAP.

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

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

3.3.3

DL(SAP) -address

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

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

3.3.4

(individual) DLSAP-address

DL-address that designates only one DLSAP within the extended link

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

3.3.5

Data-link connection endpoint address (DLCEP-address)

DL-address that designates either:

- one peer DL-connection-end-point;
- 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.

3.3.6

Frame check sequence (FCS) error

error that occurs when the computed frame check sequence value after reception of all the octets in a data link protocol data unit (DLPDU) does not match the expected residual

3.3.7

frame

synonym for DLPDU

3.3.8

network management

management functions and services that perform network initialization, configuration, and error handling

3.3.9

protocol

convention on the data formats, time sequences, and error correction for data exchange in communication systems

3.3.10

receiving DLS-user

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

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

3.3.11

sending DLS-user

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

3.3.12

device

single DLE as it appears on one local link

3.3.13

DL– entity identifier

address that designates the (single) DLE associated with a single device on a specific local link

3.3.14

device unique identification

unique 8 octet identification to identify a Type 21 device in a network. This ID is a combination of a 6 octet ISO/IEC 8802-3:2000 MAC address and 2 octet DL-address

3.3.15

ring

active network where each node is connected in series to two other devices

NOTE A ring may also be referred to as a loop.

3.3.16

linear topology

topology where the devices are connected in series, with two devices each connected to only one other device, and all others each connected to two other devices, for example, connected in a line

3.3.17

R-port

port in a communication device that is part of a ring structure

3.3.18

real-time

ability of a system to provide a required result in a bounded time

3.3.19**real-time communication**

transfer of data in real-time

3.3.20**Real-time Ethernet (RTE)**

ISO/IEC 8802-3:2000 based network that includes real-time communication

NOTE 1 Other communications can be supported, providing that the real-time communication is not compromised.

NOTE 2 This definition is based on, but not limited to, ISO/IEC 8802-3:2000. It could be applicable to other IEEE802 specifications, e.g., IEEE802.11.

3.3.21**RTE end device**

device with at least one active RTE port

3.3.22**RTE port**

media access control (MAC) sublayer point where an RTE is attached to a local area network (LAN)

NOTE This definition is derived from that of bridge port in ISO/IEC 10038: 1993, as applied to local MAC bridges.

3.3.23**switched network**

network also containing switches

NOTE Switched network means that the network is based on IEEE802.1D and IEEE802.1Q with MAC bridges and priority operations.

3.3.24**link**

transmission path between two adjacent nodes [derived from ISO/IEC 11801]

3.4 Symbols and abbreviations**3.4.1 Common symbols and abbreviations**

DL	data link (used as a prefix or adjective)
DLC	data link connection
DLCEP	data link connection endpoint
DLE	data link entity (the local active instance of the DLL)
DLL	data link layer
DLPDU	data link protocol data unit
DLPM	data link protocol machine
DLM	data link management
DLME	data link management entity (the local active instance of DLM)
DLMS	data link management service
DLS	data link service
DLSAP	data link service-access-point
DLSDU	data link service-data-unit
FIFO	first-in, first-out (queuing method)
NMT	network management
OSI	Open Systems Interconnection

Ph-	physical layer (as a prefix)
PHY	physical interface transceiver
PhL	physical layer
RTE	Real-time Ethernet
IEC	International Electrotechnical Commission
IP	Internet Protocol (see RFC 791)
ISO	International Organization for Standardization
MAC	media access control
NRT	non-real-time
PDU	protocol data unit
SAP	service access point
RT	real-time
TCP	Transmission Control Protocol (see RFC 793)
UDP	User Datagram Protocol (see RFC 768)

3.4.2 Type 21: Additional symbols and abbreviations

EFR	extremely fast recovery
GD	general device
LNМ	line network manager
PO	power on
PnP	plug and play
RNM	ring network manager
RNMP	primary ring network manager
RNMS	secondary ring network manager
RNAC	ring network auto configuration
UID	device unique identification
Type 21 NMIB	Type 21 network management information base

4 Overview of the data-link protocol

4.1 General

Type 21 extends Ethernet according to the ISO/IEC 8802-3:2000 standard with mechanisms to transfer data with predictable timing demands typical of high-performance automation. It does not change the basic principles of the Ethernet standard ISO/IEC 8802-3:2000 but extends it toward RTE. Thus, it is possible to continue to use standard Ethernet hardware, infrastructure components, or test and measurement equipment, such as network analyzers.

4.2 Overview of medium access control

A Type 21 device requires an integrated switch with two ports (ring ports) connected to the ring. A Type 21 network system is constructed with full-duplex, collision-free Ethernet switching devices as a ring or a line network. Type 21 guarantees collision-free data transmission between two devices linked by a full-duplex Ethernet connection. Thus, the Type 21 data link layer provides reliable, transparent and collision-free data transmission among DLS-users.

A Type 21 connection provides collision-free, full-duplex data communication between two devices. Therefore, a Type 21 device can transmit a frame at any time without restriction of media access rights. Type 21 data link layer does not restrict the scheduling method to use for data link resources, but each device can be individually scheduled by the application program so that Type 21 can be applied flexibly in various applications.

A Type 21 frame is delivered to the destination device in one of the following two ways:

- A frame is initiated by a source device and directly transmitted to the neighboring destination device;
- A frame is initiated by a source device and forwarded to the destination device by intermediate devices.

When a frame is forwarded by the intermediate device, the frame forwarding procedure is processed by the internal hardware switch to minimize the processing time, and it does not affect the performance of Type 21 DLL.

4.3 Service assumed from the physical layer

This standard describes the assumed physical layer service (PhS) and the constraints used by the DLE. The physical service is assumed to provide the following service primitives specified by ISO/IEC 8802-3:2000, Clause 2.

The assumed primitives of PhS are MA-DATA.request and MA-DATA.indication.

The temporal relationship of the primitives is shown in Figure 2.

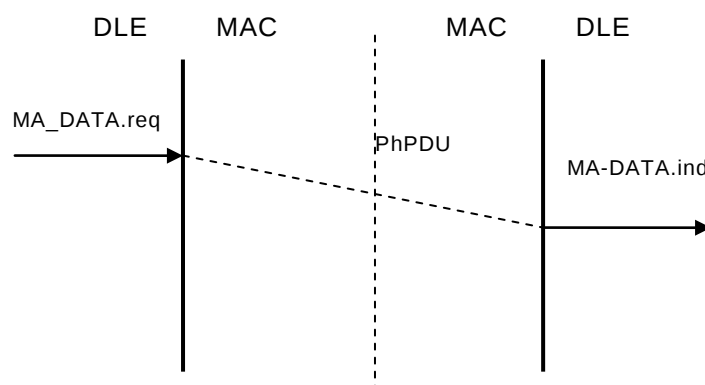


Figure 2 – Interaction of PhS primitives with DLE

The MA-DATA request primitive defines the transfer of data from a MAC client entity to a single peer entity or multiple peer entities in the case of group addresses.

The MA-DATA indication primitive defines the transfer of data from the MAC sublayer entity (through the optional MAC control sublayer, if implemented) to the MAC client entity or entities in the case of group addresses.

4.4 DLL architecture

4.4.1 General

The Type 21 DLL provides reliable and efficient higher-level data transfer service using a full-duplex ring or line topology without any specific access control scheme. The Type 21 DLL is modeled as a combination of control components of the data link protocol machine (DLPM) and the DLL management interface.

The DLPM transmits the data link service data unit (DLSDU) generated by the local DLS-user to the appropriate R-port according to the data link service policy. The DLPM also examines the received frame and delivers the received DLPDU to the appropriate DLS-user.

The DLL management interface provides DLL management functions to maintain the network management information base (NMIB). The NMIB includes local device information, network information and the path table. The DLL is comprised of the components listed in Table 1.

Table 1 – DLL components

Components	Description
DL-protocol machine (DLPM)	Transmits the DLSDU received from local DLS-user through the real-time queue (RT-queue) or non-real-time queue (NRT-queue). Examines the received frame and delivers the DLPDU to the appropriate DLS-user.
DLL management interface (DLM-interface)	Holds the station management variables that belong to the DLL, and manages synchronized changes of the link parameters.

The internal arrangement of these components, and their interfaces, are shown in Figure 3. The arrowheads illustrate the primary direction of the flow of data and control.

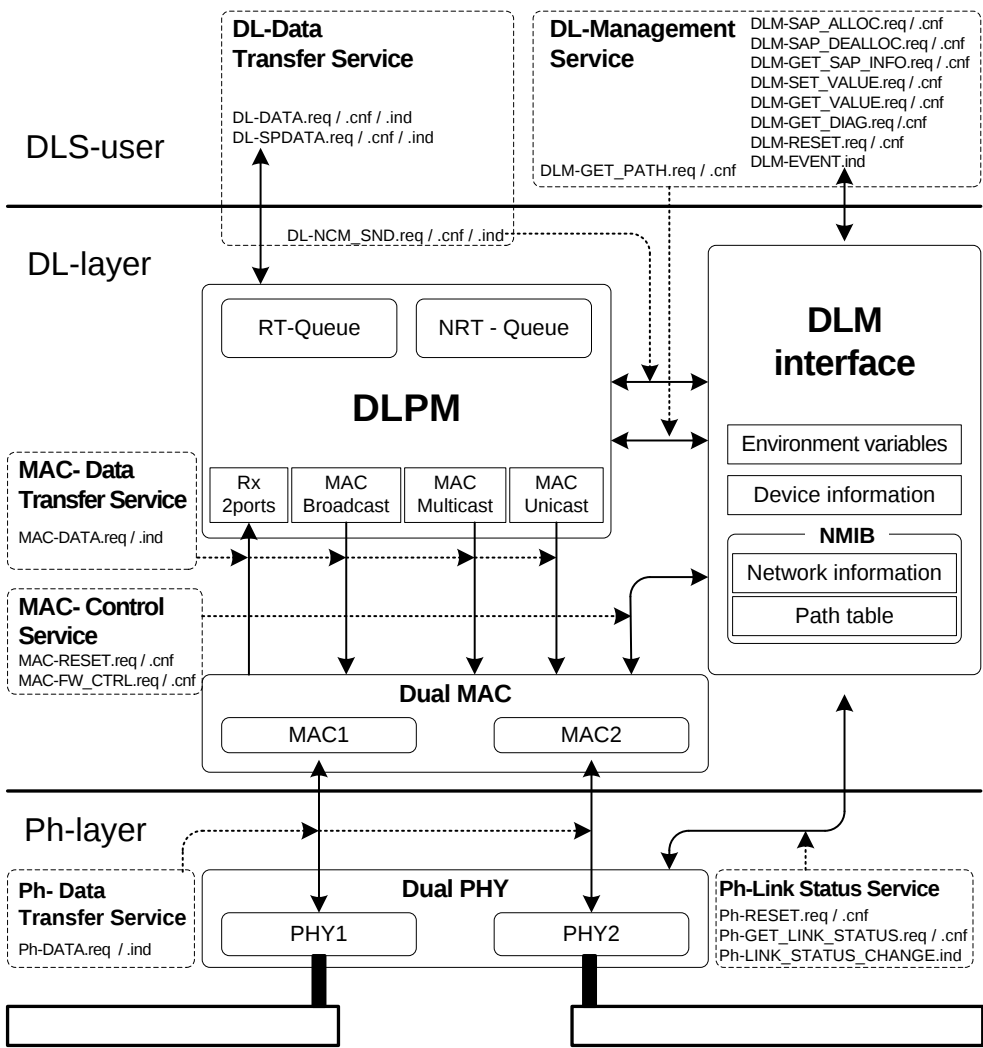


Figure 3 – Data-link layer architecture

4.4.2 DLL management (DLM) interface support function

DLM is one of the key Type 21 features that enables the Plug and Play (PnP), Network Auto Configuration (NAC), and Extremely Fast Recovery (EFR) functions. DLM spontaneously maintains and shares local device information and network-related information with every device on the network.

- Type 21 NMIB management

The DLM interface maintains the NMIB including the local device information that indicates the physical and logical status of the device, the network information that indicates the current network configuration status, and the path table that indicates the path information to the other devices and their profiles.

- Plug and Play (PnP)

When a device joins the existing network, it is automatically detected and configured without extra manual configuration of parameters so that the new device can communicate directly with the other devices on the network.

- Network auto configuration (NAC)

Type 21 supports ring and line network topologies. When the network is changed from ring to line or from line to ring topology, the change is automatically detected and broadcast to every device on the network, and then the network information and path table entries are automatically updated.

- Extremely fast recovery (EFR)

When the network is changed from ring to line topology, network information and path information are automatically updated within 10 ms.

- System diagnostic services

The DLM interface also provides system diagnostic service.

4.5 Data type

4.5.1 General

This standard describes the basic Type 21 data types. The data types described in this standard are only the normative references to represent Type 21 data type formats. The implementation issues are not covered in this standard.

4.5.2 Boolean

Data of basic data type BOOLEAN can have the values TRUE or FALSE. The values are represented as bit sequences of length 1. The value TRUE is represented by a 1 and the value FALSE is represented by a 0.

4.5.3 Unsigned integer

Data of basic data type UNSIGNED n have values of non-negative integers. The value range is $0, \dots, 2^{n-1}$. The data are represented as bit sequences of length n . The bit sequence

$$b = b_0 \sim b_{n-1}$$

is assigned value defined by

$$\text{UNSIGNED}n = b_0 2^0 + b_1 2^1 + \dots + b_{n-1} 2^{n-1}$$

The bit sequence starts on the right with the least significant octet as shown in Table 2.

Table 2 – UNSIGNEDn data type

Octet number	1	2	3	4	5	6	7	8
UNSIGNED8	$b_0..b_7$							
UNSIGNED16	$b_0..b_7$	$b_8..b_{15}$						
UNSIGNED24	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$					
UNSIGNED32	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$				
UNSIGNED40	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$			
UNSIGNED48	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$		
UNSIGNED56	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$	$b_{48}..b_{55}$	
UNSIGNED64	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$	$b_{48}..b_{55}$	$b_{56}..b_{63}$

4.5.4 Signed integer

Data of basic data type INTEGERn has values of integers. The value range is from -2^{n-1} to 2^{n-1} . The data are represented as bit sequences of length n . The bit sequence

$$b = b_0 \sim b_{n-1}$$

is assigned value defined by

$$\text{INTEGERn} = b_0 2^0 + b_1 2^1 + \dots + b_{n-2} 2^{n-2}, \text{ when } b_{n-1} \text{ is 0, and}$$

$$\text{INTEGERn} = -\text{Complement}(b) - 1, \text{ when } b_{n-1} \text{ is 1,}$$

The bit sequence starts on the right with the least significant octet as shown in Table 3.

Table 3 – INTEGERn data type

Octet number	1	2	3	4	5	6	7	8
INTEGER8	$b_0..b_7$							
INTEGER16	$b_0..b_7$	$b_8..b_{15}$						
INTEGER24	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$					
INTEGER32	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$				
INTEGER40	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$			
INTEGER48	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$		
INTEGER56	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$	$b_{48}..b_{55}$	
INTEGER64	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$	$b_{48}..b_{55}$	$b_{56}..b_{63}$

4.5.5 Octet String

The data type OCTET_STRINGn is defined below. The n represents the octet length of the octet string.

ARRAY[n] of UNSIGNED8 OCTET_STRINGn

4.5.6 Visible String

The data type VISIBLE_STRINGn is defined below. The admissible values of data of type VISIBLE_CHAR are 0x00 and the range 0x20–0x7E. The data are interpreted as 7-bit coded characters. The n indicates the octet length of the visible string.

UNSIGNED8 VISIBLE_CHAR

ARRAY[n] of VISIBLE_CHAR VISIBLE_STRINGn

There is no 0x00 necessary to terminate the string.

4.5.7 Time of day

The data type TIMEOFDAY is defined below. TIMEOFDAY consists of UNSIGNED16 date counting and UNSIGNED32 millisecond fields.

TIMEOFDAY

UNSIGNED32 milliseconds: the count from time 00:00 in milliseconds.

UNSIGNED16 date count: the number of days from 1984–01–01.

4.6 Local parameters and variables

4.6.1 General

This standard uses DLS-user request parameters P(...) and local variables V(...) as a means of clarifying the effects of certain actions and the conditions under which those actions are valid. The standard also uses local timers T(...) as a means of monitoring the actions of the

distributed DLS-provider and of ensuring a local DLE response to the absence of those actions. The standard uses 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:

- a) All variables shall be initialized to their default value, or to their minimum permitted value if no default is specified;
- b) All counters shall be initialized to zero;
- c) All timers shall be initialized to inactive.

DLM may change the values of configuration variables.

Local parameters and variables include DLE configuration parameters, local device information, and NMIB variables. The DLE operational condition is stored in DLE configuration parameters and the DLE configuration parameters are configured locally or remotely according to the application. Local data link information, such as DL- entity identifier and the status of each R-port are stored in the local device information. NMIB includes network information and the path table. Network topology and the network-related variables are stored in the network information, and the device profile and path information of the other devices on the network are summarized in the path table.

4.6.2 DLE configuration parameters

4.6.2.1 General

These parameters are required to configure the local DLE operation, and they are managed by DLM. Every device on the same network segment has the same DLE configuration parameters. DLM-GET_VALUE service and DLM-SET_VALUE service are used to read or write the DLE configuration parameters. Table 4 shows the list of DLE configuration parameters.

Table 4 – DLE configuration parameters

Parameter	Data type	Default value	Description
Max DL- entity identifier	UNSIGNED16	255	Maximum DL- entity identifier 256 to 65 535: Reserved
DLPM scheduling policy	UNSIGNED8	0	0: First-In, First-Out 1: Fixed priority 2: to 255: Reserved

4.6.2.2 P(MAX_ADDR): Maximum DL- entity identifier

This variable holds the maximum device address and is set by the DLM. The range of this variable is 1 to 255. The default value of this variable is 255. This variable also indicates the maximum number of path table entries in the NMIB. The values in the range 256 to 65 535 are reserved.

4.6.2.3 P(DLPMSP): data link protocol machine scheduling policy

This variable holds the scheduling policy of the local DLPM, which dictates how to serve the concurrent DLSDUs in the RT-queue. The NRT-queue is not scheduled by the DLPMSP. DLPMSP can have one of the following values:

- a) 0: first-in, first-out. The first received frame is served first. In this case, the message priority in the DLSDU is ignored;

- b) 1: Fixed priority. The DLSDU with high priority is served first. If many DLSDUs are stacked with the same priority, the frame received first is served first;
- c) 2 to 255: reserved.

4.6.3 Queues to support data transfer

4.6.3.1 General

When a DLS-user generates a DLSDU to be transmitted by the DLPM, the DLSDU is encapsulated by the DLPDU. The DLPDU is then stored in the RT-queue or the NRT-queue according to its application and service type. Table 5 shows the queue list for Type 21 data transfer.

Table 5 – Queues to support data transfer

Parameter	Data type	Default value	Description
RT-queue	—	—	Transmit queue to store the real-time data
NRT-queue	—	—	Transmit queue to store the non-real-time data

4.6.3.2 Q(RTQ): RT-queue

The RT-queue stores the real-time DLPDUs generated by a DLS-user to be transmitted by the DLPM. The size of the RT-queue is not restricted in this standard and it is considered a detail of the local implementation. The RT-queue is scheduled by the DLPM.

4.6.3.3 Q(NRTQ): NRT-queue

The NRT-queue stores non-real-time DLPDUs generated by a DLS-user. The size of the NRT-queue is not restricted in this standard and it is considered a detail of the local implementation. Messages in the RT-queue are transmitted first. Messages in the NRT-queue are transmitted only when the RT-queue is empty.

4.6.4 Variables to support SAP management

4.6.4.1 General

Allocation and de-allocation of the data link service access point (DLSAP) is managed by the DLM. When a frame is received, the DLPM examines the destination service access point (DSAP) in the DLSDU. If the DSAP is already allocated to a DLS-user, the DLM returns the appropriate DLS-user ID equivalent for the received DSAP address, and the DLPM delivers the received DLSDU to the DLS-user. If the service access point (SAP) is not allocated and no appropriate DLS-user ID is found in the DLM, the received DLSDU is discarded by the DLPM. Therefore, to receive a DLSDU from a certain peer DLS-user, a DLS-user shall first obtain a SAP allocation using the DLM-SAP_ALLOC service. Once the SAP is allocated to a DLS-user, it is used to send and receive data until the SAP is deallocated and returned to the DLM. The deallocated SAP can be used again after reallocation. The SAP address and its appropriate DLS-user ID are stored together and maintained by the DLM. The maximum number of SAP management items is 65 535 but the method to allocate and de-allocate the SAP address is not restricted in this standard. Table 6 shows the list of SAP management variables.

Table 6 – Variables to support SAP management

Parameter	Data type	Default value	Description
SAP	UNSIGNED16	—	Service access point
DLS-user ID	UNSIGNED32	—	Numeric identification of the DLS-user that owns the SAP allocation

4.6.4.2 V(SAP): SAP

This variable holds the SAP. The value for this variable is in the range 0 to 65 535.

4.6.4.3 V(DLS_USER_ID): DLS-user ID

This variable holds the 4-octet numeric identification of the DLS-user who owns the SAP allocation.

4.6.5 Variables to support local device information management**4.6.5.1 General**

To maintain the network topology, every device manages a device database including the local device information and the other device information. Table 7 shows the list of device information management variables.

Table 7 – Variables to support device information management

Parameter	Data type	Default value	Description
DL– entity identifier	UNSIGNED16	INVALID_ADDR	Local DL– entity identifier
Device flags	UNSIGNED16	0	Local device flags
Device type	UNSIGNED16	0	Local device type
Hop count	UNSIGNED16	0	Hop count
Device UID	UNSIGNED64	INVALID_UID	Local device unique ID
Device UID for R-port1	UNSIGNED64	INVALID_UID	Device unique ID connected through R-port1
Device UID for R-port2	UNSIGNED64	INVALID_UID	Device unique ID connected through R-port2
MAC address	UNSIGNED48	0	Local device MAC address
Reserved0	UNSIGNED16	0	Reserved
Port information	UNSIGNED16	0	Local device port information
Device state	UNSIGNED8	0	DLM state
Protocol version	UNSIGNED8	0	Local device protocol version
Device description	VISIBLE_STRING[16]	“ ”	Device description string
Reserved1	UNSIGNED32	0	Reserved

4.6.5.2 V(DL_ADDR): DL–entity identifier

This variable holds the DL–entity identifier that designates the (single) DL–entity associated with a single device on a specific local link whose value is constrained to the range 0 to 255. The DL– entity identifier may be provided by hardware settings (for example, rotary switch) or set by software. The DL– entity identifier is defined in Table 8.

Table 8 – DL–entity identifier

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	Read	0 to 255: DL– entity identifier 256 to 65 535: reserved

4.6.5.3 V(DEV_FLAG): Device Flag

This variable holds the flags for events that occurred in a local device. When the local DL–entity identifier collision flag is set, EVENT_THIS_ADDR_COLLISION event is generated by the DLM, and then the local DL– entity identifier collision flag is cleared. When the DLM state change flag is set, EVENT_DEV_STATE_CHG is generated by the DLM, and then the DLM state change flag is cleared. Type 21 device flags and event flags are listed in Table 9.

Table 9 – Device Flags

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	Read	0x00: normal 0x01: collision 0x02: changed 0x03: collision and changed All others reserved

4.6.5.4 V(DLM_STATE): DLM state

This variable holds the DLM state. DLM state is defined as shown in Table 10.

Table 10 – DLM state

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED8	Read/Write	Read	0: INVALID_DLM_STATE 1: standalone state (SA) 2: line network manager state (LNM) 3: general Device state (GD) 4: primary ring network manager state (RNMP) 5: secondary ring network manager state (RNMS) 6 to 255: Reserved

4.6.5.5 V(DEV_UID): Device UID

This variable holds the unique 8-octet identification that identifies a Type 21 device in a network. It is a combination of the 6-octet ISO/IEC 8802-3:2000 MAC address and the 2-octet DL– entity identifier. The Device UID is defined as shown in Table 11.

Table 11 – Device Unique Identification

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED64	Bit 0 – 15	Read/Write	Read	0 to 255: DL– entity identifier 256 to 65 535: Reserved
	Bit 16 – 63	Read/Write	Read	ISO/IEC 8802-3:2000 MAC Address

4.6.5.6 V(DEV_UID_RP1): Device UID for R-port1

This variable holds the UID of the device that is linked through the R-port1. The Device UID for R-port1 is defined as shown in Table 12.

Table 12 – Unique identification of device connected to R-port1

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED64	Bit 0 – 15	Read/Write	Read	0 to 255: DL– entity identifier 256 to 65 535: Reserved
	Bit 16 – 63	Read/Write	Read	ISO/IEC 8802-3:2000 MAC Address

4.6.5.7 V(DEV_UID_RP2): Device UID for R-port2

This variable holds the UID of the device that is linked through the R-port2. The Device UID for R-port2 is defined as shown in Table 13.

Table 13 – Unique identification of device connected to R-port2

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED64	Bit 0 – 15	Read/Write	Read	0 to 255: DL– entity identifier 256 to 65 535: Reserved
	Bit 16 – 63	Read/Write	Read	ISO/IEC 8802-3:2000 MAC Address

4.6.5.8 V(MAC_ADDR): MAC address

This variable holds 6-octet ISO/IEC 8802-3:2000 Ethernet MAC address of local device. As a Type 21 device has two Ethernet MAC ports, both MAC addresses should be identical. The MAC address is defined as shown in Table 14.

Table 14 – MAC address

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED48	Read/Write	Read	ISO/IEC 8802-3:2000 MAC Address

4.6.5.9 V(PORT_INFO): Port information

This variable holds the port information for each R-port. It is defined as shown in Table 15.

Table 15 – Port information

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Bit 0	Read/Write	Read	R-port1 link down
	Bit 1	Read/Write	Read	Received NCM_FAMILY_RES message from R-port1
	Bit 2	Read/Write	Read	Waiting for NCM_ADV_THIS message from R-port1
	Bit 3	Read/Write	Read	Waiting for NCM_MEDIA_LINKED message from R-port1
	Bit 4	Read/Write	Read	R-port1 state confirm
	Bit 5 – 7	Read/Write	Read	Reserved
	Bit 8	Read/Write	Read	R-port2 link down
	Bit 9	Read/Write	Read	Received NCM_FAMILY_RES message from R-port2
	Bit 10	Read/Write	Read	Waiting for NCM_ADV_THIS message from R-port2
	Bit 11	Read/Write	Read	Waiting for NCM_MEDIA_LINKED message from R-port2
	Bit 12	Read/Write	Read	R-port2 state confirm
	Bit 13 – 15	Read/Write	Read	Reserved

4.6.5.10 V(PROTOCOL_VER): Protocol version

This variable holds the protocol version of local device. It is defined as shown in Table 16.

Table 16 – Protocol version

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED8	Bit 0 – 1	Read	Read	0x00: major version 1 0x01: major version 2 0x02: major version 3 0x03: major version 4
	Bit 2 – 4	Read	Read	0x00: minor version 0 0x01: minor version 1 0x02: minor version 2 0x03: minor version 3 0x04: minor version 4 0x05: minor version 5 0x06: minor version 6 0x07: minor version 7
	Bit 5 – 7	—	—	Reserved

4.6.5.11 V(DEV_TYPE): Device type

This variable holds the local device type that represents the general function of the device. The value of this variable is defined as shown in Table 17.

Table 17 – Device type

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Bit 0 – 7	Read	Read/Write	0 to 255: general device type 0: invalid device type 1: programmable logic controller (PLC) 2: motion controller 3: human-machine interface (HMI) 4: industrial personal computer 5: inverter 6: simple I/O 7 to 255: reserved
	Bit 8 – 15	Read	Read/Write	0 to 255: application specific device type

4.6.5.12 V(DEV_DESC): Device description

This variable contains a description of the local device. The maximum length of this variable is 16 octets. It is defined as shown in Table 18.

Table 18 – Device description

Data type	Access type DLM	Access type DLS-user	Value/Description
VISIBLE_STRING[16]	—	Read/Write	Any string defined by a DLS-user set using the set DLL configuration service

4.6.5.13 V(HOP_CNT): Hop count

This variable holds the count of the number of devices between two devices. When the DLM receives NCM_ADV_THIS or NCM_LA DLPDU, the DLM saves the received hop count value in this variable, then increments the hop counts in the received frame by 1 and transmits the frame through the other R-port. In this way, each device builds its own path table with a hop count for R-port1 and a hop count for R-port2. This variable is defined as shown in Table 19.

Table 19 – Hop count

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	—	0 to 255: Hop count 256 to 65 535: reserved

4.6.6 Variables and counter to support network information management**4.6.6.1 General**

Network information is managed automatically by the DLM. Network information variables and counters are summarized in Table 20.

Table 20 – Variables to support managing network information

Parameter	Data type	Default value	Description
Topology	UNSIGNED8	NET_TPG_SA	Network topology
Collision count	UNSIGNED8	0	DL–entity identifier collision counter between remote devices
Device count	UNSIGNED16	1	Device counter for the network segment
Topology change count	UNSIGNED16	0	Network topology change counter
Network flags	UNSIGNED16	0	Network event flags
Last topology change time	TIMEOFDAY	0	Date and time when the network topology last was changed.
Reserved	UNSIGNED16	0	Reserved
RNMP device UID	UNSIGNED64	INVALID_UID	UID of the RNMP device
RNMS device UID	UNSIGNED64	INVALID_UID	UID of the RNMS device
LNM device UID for R-port1	UNSIGNED64	INVALID_UID	UID of the LNM device in R-port1 direction
LNM device UID for R-port2	UNSIGNED64	INVALID_UID	UID of the LNM device in R-port2 direction

4.6.6.2 V(TPG): Topology

This variable holds the type of network topology. The value of this variable is defined as shown in Table 21.

Table 21 – Topology

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED8	Read/Write	Read	Network topology 0x01: standalone 0x02: line topology 0x03: ring topology

4.6.6.3 C(COLL_CNT): Collision count

This variable holds the DL– entity identifier collision count for remote devices. The value is incremented by the DLM when a remote DL– entity identifier collision is detected and the value is decremented when the collision is cleared. This variable is defined in Table 22.

Table 22 – Collision count

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED8	Read/Write	Read	0 to 255: remote DL– entity identifier collision count

4.6.6.4 C(DEV_CNT): Device Count

This variable holds the total number of devices on the network, to a maximum of 256. It is defined as shown in Table 23.

Table 23 – Device count

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	Read	0: not used 1 to 256: device count 257 to 65 535: reserved

4.6.6.5 C(TPG_CHG_CNT): Topology change count

This variable holds the topology change count. The value is incremented by the DLM when the network is changed from ring to line or from line to ring topology. It is defined as shown in Table 24.

Table 24 – Topology change count

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	Read	0 to 65 535: Topology change count

4.6.6.6 V(TPG_CHG_TIME): Last topology change time

This variable holds the date and time when the network topology was last changed. It is defined as shown in Table 25.

Table 25 – Last topology change time

Data type	Access type DLM	Access type DLS-user	Value/Description
TIMEOFDAY	Read/Write	Read	The date and time when the network topology was last changed

4.6.6.7 V(UID_RNMP): RNMP device UID

This variable holds the device UID selected as the RNMP on the network. It is defined as shown in Table 26.

Table 26 – RNMP device UID

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED64	Bit 0 – 15	Read/Write	Read	0 to 255: DL– entity identifier 256 to 65 535: Reserved
	Bit 16 – 63	Read/Write	Read	ISO/IEC 8802-3:2000 MAC Address

4.6.6.8 V(UID_RNMS): RNMS device UID

This variable holds the UID of the device selected as the RNMS on the network. It is defined as shown in Table 27.

Table 27 – RNMS device UID

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED64	Bit 0 – 15	Read/Write	Read	0 to 255: DL– entity identifier 256 to 65 535: Reserved
	Bit 16 – 63	Read/Write	Read	ISO/IEC 8802-3:2000 MAC Address

4.6.6.9 V(UID_LNM_RP1): LNM device UID for R-port1

In a Type 21 line network, the two end devices are automatically selected as the LNM. This variable holds the UID of the device selected as the LNM in the R-port1 direction. It is defined as shown in Table 28.

Table 28 – LNM device UID for R-port1

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED64	Bit 0 – 15	Read/Write	Read	0 to 255: DL– entity identifier 256 to 65 535: Reserved
	Bit 16 – 63	Read/Write	Read	ISO/IEC 8802-3:2000 MAC Address

4.6.6.10 V(UID_LNM_RP2): LNM device UID for R-port2

In a Type 21 line network, two end devices are automatically selected as the LNM. This variable holds the UID of the device selected as the LNM in the R-port2 direction. It is defined as shown in Table 29.

Table 29 – LNM device UID for R-port2

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED64	Bit 0 – 15	Read/Write	Read	0 to 255: DL– entity identifier 256 to 65 535: Reserved
	Bit 16 – 63	Read/Write	Read	ISO/IEC 8802-3:2000 MAC Address

4.6.6.11 V(NET_FLAG): Network flags

This variable holds the event flags to notify the DLMS-user of network events. When any bit in this variable is set, the DLM generates a DLM-event indication service primitive to notify the DLMS-user of the event. After the DLM-event service has completed successfully, the designated bit in this variable is cleared. Network flags are defined in Table 30.

Table 30 – Network flags

Data type	Position	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Bit 0	Read/Write	Read	Network Topology Change Status 0x00: Normal 0x01: Network topology has changed
	Bit 1	Read/Write	Read	Network DL– entity identifier Collision Status 0x00: Normal 0x01: Network DL– entity identifier collision detected
	Bit 2	Read/Write	Read	In Device Status 0x00: Normal 0x01: New device has joined the network
	Bit 3	Read/Write	Read	Out Device Status 0x00: Normal 0x01: Device has left the network
	Bit 4 – 63	—	—	Reserved

The bit fields in the network flags are defined as follows:

a) Network Topology Change Status;

This bit is set to TRUE by the DLM when it detects that the network topology has changed. This bit is reset to FALSE when the DLMS-user has been notified using the EVENT_NET_TPG_CHG service.

b) Network DL– entity identifier Collision Status;

This bit is set to TRUE by the DLM when it detects a DL– entity identifier collision on the network. This bit is reset to FALSE when the DL– entity identifier collision is cleared.

c) In Device Status;

This bit is set to TRUE by the DLM when it detects that a new device has joined the network. This bit is reset to FALSE when the DLMS-user has been notified using the EVENT_IN_DEVICE service.

d) Out Device Status.

This bit is set to TRUE by the DLM when it detects that a device has left the network. This bit is reset to FALSE when the DLMS-user has been notified using the EVENT_OUT_DEVICE service.

4.6.7 Variables and counter to support a device path information management

4.6.7.1 General

The path table is managed by the DLM, and it is made up of table items related to the other Type 21 devices on the network. The variables and counters for a table item are defined in Table 31.

Table 31 – Variables and counter to support managing path information

Parameter	Data type	Default value	Description
DL– entity identifier	UNSIGNED16	INVALID_ADDR	DL– entity identifier
Hop count for R-port1	UNSIGNED16	INVALID_HOP_CNT	Hop count in R-port1 direction
Hop count for R-port2	UNSIGNED16	INVALID_HOP_CNT	Hop count in R-port2 direction
Preferred R-port	UNSIGNED8	INVALID_R_PORT	R-port with the smaller hop count value to the peer device.
Destination R-port	UNSIGNED8	INVALID_R_PORT	Selected R-port for sending a frame to the destination DL– entity identifier.
Device state	UNSIGNED8	0	Peer device's DLM State
MAC address	UNSIGNED48	0	Peer device's ISO/IEC 8802-3:2000 MAC address
Port information	UNSIGNED16	0	Peer device's local R-port information
Protocol version	UNSIGNED8	0	Peer device's protocol version
Device type	UNSIGNED16	0	Peer device's application device type
Device description	VISIBLE_STRING[16]	“ ”	Peer device's description
Device UID	UNSIGNED64	INVALID_UID	Peer device's Device UID
Device UID for R-port1	UNSIGNED64	INVALID_UID	Device UID of the device connected through the peer device's R-port1
Device UID for R-port2	UNSIGNED64	INVALID_UID	Device UID of the device connected through the peer device's R-port2
In net count	UNSIGNED16	0	The number of times that the peer device has joined in the network.
In net time	TIMEOFDAY	0	The date and time when the peer device last joined the network.
Out net count	UNSIGNED16	0	The number of times that the peer device has been disconnected from the network.
Out net time	TIMEOFDAY	0	The date and time when the peer device was last disconnected from the network

4.6.7.2 V(path-DL_ADDR): DL– entity identifier

See 4.6.5.2.

4.6.7.3 C(path-HOP_CNT_RP1): Hop count for R-port1

This variable indicates the frame forwarding counts for sending a frame from the local device to the peer device through the R-port1. It is defined as shown in Table 32.

Table 32 – Hop count for R-port1 direction

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	Read	0 to 255: Hop count for R-port1 256 to 65 535: Reserved

4.6.7.4 C(path-HOP_CNT_RP2): Hop count for R-port2

This variable indicates the frame forwarding counts for sending a frame from the local device to the peer device through the R-port2. It is defined as shown in Table 33.

Table 33 – Hop count for R-port2 direction

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	Read	0 to 255: hop count for R-port2 256 to 65 535: Reserved

4.6.7.5 V(path-PREFER_RP): Preferred R-port

This variable holds the preferred R-port for sending a frame from the local device to the peer device without regard for the RNMP or RNMS. This variable is determined as the R-port that has the smaller hop count value for the peer device. If the R-port1 hop count and R-port2 hop count have the same value, R-port1 is selected as the preferred R-port. This variable is defined as shown Table 34.

Table 34 – Preferred R-port

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED8	Read/Write	Read	0: Invalid 1: R-port1 2: R-port2 3 to 255: Reserved

4.6.7.6 V(path-DST_RP): Destination R-port

This variable holds the destination R-port for sending a frame from the local device to the peer device. In a line network, this variable has the same value as the Preferred R-port. However, in a ring network, this variable is determined based on the RNMP and RNMS position, because the preferred path may be blocked by the RNMP or RNMS. In this case, the destination R-port is selected as the other R-port. It is defined as shown in Table 35.

Table 35 – Destination R-port

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED8	Read/Write	Read	0: Invalid 1: R-port1 2: R-port2 3 to 255: Reserved

4.6.7.7 V(path-DEV_STATE): Device State

See 4.6.5.4.

4.6.7.8 V(path-MAC_ADDR): MAC address

See 4.6.5.8.

4.6.7.9 V(path-PORT_INFO): Port information

See 4.6.5.9.

4.6.7.10 V(path-PROTOCOL_VER): Protocol Version

See 4.6.5.10.

4.6.7.11 V(path-DEV_TYPE): Device type

See 4.6.5.11.

4.6.7.12 V(path-DEV_DESC): Device Description

See 4.6.5.12.

4.6.7.13 V(path-DEV_UID): Device UID

See 4.6.5.5.

4.6.7.14 V(path-DEV_UID_RP1): Device UID for R-port1

See 4.6.5.6.

4.6.7.15 V(path-DEV_UID_RP2): Device UID for R-port2

See 4.6.5.7.

4.6.7.16 C(path-IN_NET_CNT): In net count

This variable holds the number of times that the peer device has joined the network. When a line network is merged into an existing network, the variables for the newly joined devices are incremented together. This variable is defined as shown in Table 36.

Table 36 – In net count

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	Read	0 to 65 535: the number of times that the peer device has joined the network

4.6.7.17 V(path-IN_NET_TIME): In net time

This variable holds the date and time when the peer device last joined in the network. This variable is defined as shown in Table 37.

Table 37 – In net time

Data type	Access type DLM	Access type DLS-user	Value/Description
TIMEOFDAY	Read/Write	Read	The date and time when the device last joined in the network

4.6.7.18 C(path-OUT_NET_CNT): Out net count

This variable holds the number of times that the peer device has been disconnected from the network. When a device or a group of devices are disconnected from the network, the variables for the disconnected devices are incremented together. This variable is defined as shown in Table 38.

Table 38 – Out net count

Data type	Access type DLM	Access type DLS-user	Value/Description
UNSIGNED16	Read/Write	Read	0 to 65 535: the number of times that the device has been disconnected from the network

4.6.7.19 V(path-OUT_NET_TIME): Out net time

This variable holds the date and time when the device was last disconnected from the network. This variable is defined as shown in Table 39.

Table 39 – Out net time

Data type	Access type DLM	Access type DLS-user	Value/Description
TIMEOFDAY	Read/Write	Read	The date and time when the device was last disconnected from the network

4.6.8 Variables, counters, timers, and queues to support path table management

4.6.8.1 Path table

The path table is composed of items related to the other devices on the network. It is managed by the DLM in the form of an array table filled with the path information of each device (see 4.6.7). The maximum size of the path table is a function of MAX_ADDR (see 4.6.2.2) as follows:

Path table: Array[n] of device's path information, $n = \text{MAX_ADDR} + 1$

5 General structure and encoding

5.1 Overview

The DLL and its procedures are necessary to provide services to the DLS-user using the services available from the physical layer. This clause describes the structure and semantics of the management application protocol data unit (MAPDU), the DLPDU, and the procedure commonly used in this standard. This portion is identical to and fully compliant with the ISO/IEC 8802-3:2000 standard.

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

5.2 MAPDU structure and encoding

The local MAC sublayer uses the service primitives provided by the physical layer service (PLS) sublayer specified by ISO/IEC 8802-3:2000 Clause 2. These service primitives provided by the PLS sublayer are mandatory:

- a) MA-DATA request;

b) MA-DATA indication.

5.3 Common MAC frame structure, encoding and elements of procedure

5.3.1 MAC frame structure

5.3.1.1 MAC frame format for the Type 21 DLPDU

The DLPDU for the Type 21 is encapsulated in the data field of a MAC frame as specified by ISO/IEC 8802-3:2000, Clause 3. The value of the Length/Type field is 88FE_H, which is authorized and registered as the protocol identification number by the IEEE Registration Authority, to identify a Type 21 fieldbus frame. Figure 4 shows the Type 21 DLPDU structure.

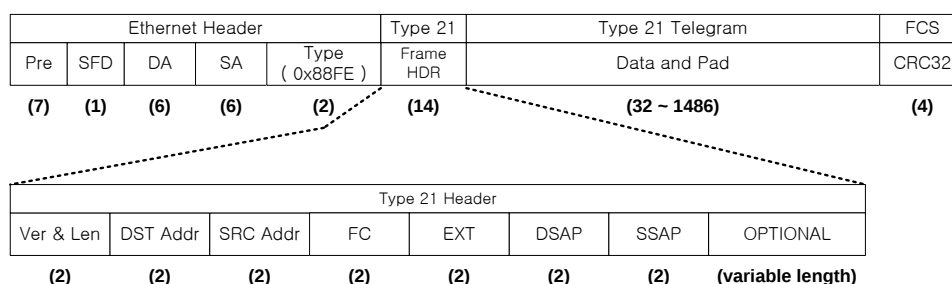


Figure 4 – Common MAC frame format for Type 21 DLPDU

5.3.1.2 MAC frame format for Type 21 fieldbus sporadic DLPDU

The MAC frame format used for Type 21 fieldbus sporadic data transmission is identical to the frame format of Ethernet V2.0 specified by ISO/IEC 8802-3:2000, Clause 3 “Media access control frame structure,” and the value of the Length/Type field is anything other than 0x88FE. Figure 5 shows the frame format for a Type 21 fieldbus sporadic DLPDU.

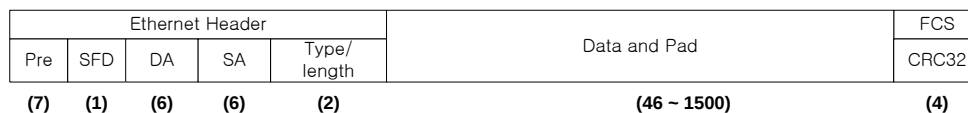


Figure 5 – MAC frame format for other protocols

5.3.2 Elements of the MAC frame

5.3.2.1 General

The elements of the MAC frame are the preamble, the start frame delimiter, the destination MAC address, the source MAC address, the length/type code, and the frame check sequence (FCS), all as specified by ISO/IEC 8802-3:2000, Clause 3.

5.3.2.2 Preamble field

The preamble of MAC frame is identical to ISO/IEC 8802-3:2000, Clause 3. This is a 7-octet field that is used to allow the physical signalling portion of the circuitry to reach its steady-state synchronization with the receiving frame timing. The preamble pattern is:

“10101010 10101010 10101010 10101010 10101010 10101010 10101010”

The bits are transmitted in order from left to right. The nature of the pattern is such that for Manchester encoding, it appears as a periodic waveform on the medium that enables bit synchronization. It should be noted that the preamble ends with a “0”.

5.3.2.3 Start Frame Delimiter

The Start Frame Delimiter (SFD) is identical to ISO/IEC 8802-3:2000, Clause 3. The SFD field is the bit pattern sequence “10101011.” It immediately follows the preamble pattern and indicates the start of a frame.

5.3.2.4 Destination MAC Address field

The Destination MAC Address field is identical to ISO/IEC 8802-3:2000, Clause 3. It specifies the device(s) for which the frame is intended, and may be an individual or multicast (including broadcast) address. The destination MAC address is set to the corresponding DL– entity identifier by the DLM. Type 21 also defines a special MAC address, 00-E0-91-02-05-99 (NCM_MAC_ADDR) for sharing network management information using the DLM services. Every message received through the NCM_MAC_ADDR is delivered to the DLM to update the network management information. The message is not forwarded by the MAC layer but the message is examined and forwarded by the DLM.

5.3.2.5 Source MAC address field

The Source MAC Address field is identical to ISO/IEC 8802-3:2000, Clause 3. This field specifies the device sending the frame and is not interpreted by the DLE or the CSMA/CD MAC sublayer.

5.3.2.6 Length/type field

The Length/type field is identical to ISO/IEC 8802-3:2000, Clause 3 “Media access control frame structure.” To be identified as a Type 21 frame, the value of the Length/type field is set to 0x88FE, which is authorized and registered as the protocol identification number for RTE-Type 21 by the IEEE Registration Authority. Every frame with a value other than 0x88FE is identical to the frame in ISO/IEC 8802-3:2000, Clause 3, and is processed as a Type 21 fieldbus sporadic data frame.

5.3.2.7 Frame check sequence

The FCS field is identical to ISO/IEC 8802-3:2000, Clause 3.

5.3.3 Elements of the Type 21 DLPDU

5.3.3.1 Version and Length

This field stores the protocol version and the length of a Type 21 telegram or data field. This Version and Length field is specified in Figure 6. The version is represented by 2 bits for the major version and 3 bits for the minor version, and the length is given by 11 bits.

Version and Length															
Version					Length										
15	14	13	21	11	10	9	8	7	6	5	4	3	2	1	0

Figure 6 – Version and Length field

The parts of this field and permissible values are described in Table 40.

Table 40 – Version and Length

Field Name		Position	Value/Description
Length		Bit 0 – 10	Frame length including FCS field
Version	Minor	Bit 11 – 13	Type 21 Protocol minor version 0x00: minor version 0 0x01: minor version 1 0x02: minor version 2 0x03: minor version 3 0x04: minor version 4 0x05: minor version 5 0x06: minor version 6 0x07: minor version 7
	Major	Bit 14 – 15	Type 21 Protocol major version 0x00: major version 1 0x01: major version 2 0x02: major version 3 0x03: major version 4

5.3.3.2 DST_addr**5.3.3.2.1 General**

This field indicates the destination DL– entity identifier of the node to which the frame is sent. This value is represented as shown in Figure 7.

Destination DL–address															
DST_Addr															
15	14	13	21	11	10	9	8	7	6	5	4	3	2	1	0

Figure 7 – DST_addr field

The separate field and its permissible values are described in Table 41.

Table 41 – Destination DL– entity identifier

Field Name	Position	Value/Description
DST_addr	Bit 0 – 15	0xFFFF: broadcast address 0xFFFE: network control address (C_NCM_ADDR) 0xFFFD–0xFFDE: user-defined multicast address 0xFFDD: invalid address 0x0100 to 0xFFDC: reserved 0x0000 to 0x00FF: regular Type 21 DL– entity identifier

5.3.3.2.2 Broadcast address

If the destination DL– entity identifier is 0xFFFF, the destination MAC address field contains the ISO/IEC 8802-3:2000 MAC address.

5.3.3.2.3 Network control address

If the destination DL– entity identifier is 0xFFFE (C_NCM_ADDR), the destination MAC address field contains C_NCM_MAC_ADDR. However, NCM_LINK_ACTV and NCM_ADV_THIS messages are transmitted using C_NCM_ADDR as the destination DL– entity identifier.

NOTE C_NCM_ADDR cannot be accessed by the DLS-user.

5.3.3.2.4 User-defined multicast address

A user-defined multicast address is used to indicate multiple recipients. However, user-defined multicast addressing is not a mandatory feature in this standard. It is designed for use in a special application system that requires multicast communication. Therefore, user-defined multicast addressing is not interoperable between heterogeneous devices. The destination DL– entity identifier range from 0xFFFFD–0xFFDE is used to specify the user-defined multicast address. However, the method of using the user-defined multicast address is not specified in this standard, and is considered a local responsibility. This standard does not restrict the use of user-defined multicast addresses, nor is it a mandatory feature.

5.3.3.3 SRC_addr

This field indicates the source DL– entity identifier of the node from which the frame is generated. This value is represented as shown in Figure 8.

Source DL–address														
SRC_Addr														
15	14	13	21	11	10	9	8	7	6	5	4	3	2	0

Figure 8 – SRC_addr field

The separate field and its permissible values are described in Table 42.

Table 42 – Source DL– entity identifier

Field Name	Position	Value/Description
SRC_addr	Bit 0 – 15	Source DL– entity identifier

5.3.3.4 Frame Control (FC)

5.3.3.4.1 General

The Frame Control field indicates the frame control information. This value is represented as shown in Figure 9.

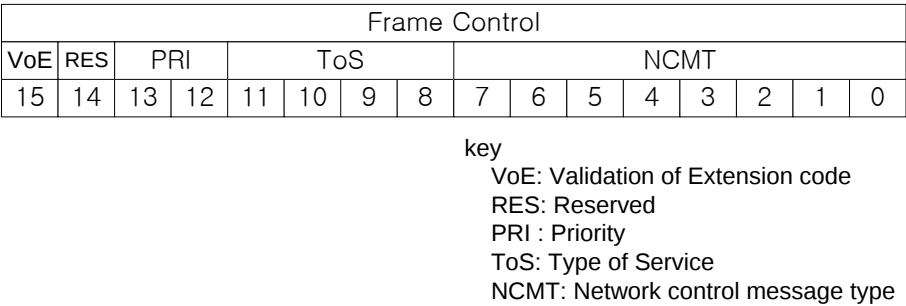


Figure 9 – Frame Control Field

The separate field and its permissible values are described in Table 43.

Table 43 – Frame control

Field Name	Position	Value/Description
Network Control Message Type (NCMT)	Bit 0 – 7	0x00: reserved 0x01: NCM_FAMILY_REQ 0x02: NCM_FAMILY_RES 0x03: NCM_MEDIA_LINKED 0x04: NCM_ADV_THIS 0x05: NCM_LINE_START 0x06: NCM_RING_START 0x07: NCM_ACK_RNMS 0x08: NCM_RETRY_RNMS 0x09 to 0xFF:reserved
Type of Service (ToS)	Bit 8 – 11	0x00: Network Control Message (NCM) 0x01: unconfirmed service request 0x02–0x0F: Reserved
Priority (PRI)	Bit 12 – 13	0x00: lowest priority ... 0x03: highest priority
Reserved	Bit 14	Reserved
Validation of Extension code (VoE)	Bit 15	0x00: EXT Code is invalid 0x01: EXT Code is valid

5.3.3.4.2 Validation of extension code (VoE)

If the frame has the extension field, VoE is set to TRUE; otherwise VoE is set to FALSE.

5.3.3.4.3 Priority

This field indicates the frame priority. This field contains the value of the message priority parameter for the DL service. The highest priority is 0x03 and the lowest is 0x00.

5.3.3.4.4 Type of service (ToS)

This field indicates the type of DL service. A value of 0x00 indicates a network control message among DLMs, and 0x01 indicates the unconfirmed service request among DLS-users.

5.3.3.4.5 Network Control Message Type (NCMT)

5.3.3.4.5.1 General

NCMT field indicates the type of network control message.

5.3.3.4.5.2 NCM_FAMILY_REQ

NCMT: 0x01

This network control message is used to ask the device newly connected through an R-port if it is a Type 21 device. This network control message is transmitted through the newly activated R-port. This message shall not be forwarded to the other port.

5.3.3.4.5.3 NCM_FAMILY_RES

NCMT: 0x02

This network control message is used to confirm whether the recipient is a Type 21 device when the recipient receives the NCM_FAMILY_REQ message from the newly linked device. This message is transmitted through the R-port used to receive the NCM_FAMILY_REQ message. This message shall not be forwarded to the other port.

5.3.3.4.5.4 NCM_MEDIA_LINKED

NCMT: 0x03

This network control message is used to indicate that a new Type 21 link has been established through the R-port. This message is transmitted through the newly activated R-port. The destination DL- entity identifier contains C_NCM_ADDR. When the DLM receives this message, the DLM increments the hop count in the frame, and forwards the frame through the other R-port. This message is discarded by the LNM or the device that generated the message.

5.3.3.4.5.5 NCM_ADV_THIS

NCMT: 0x04

This network control message is used to transmit the recipient's local device information when the recipient receives NCM_MEDIA_LINKED message from a new device on the network. This message is transmitted through the R-port that is used to receive the NCM_MEDIA_LINKED message. The destination DL- entity identifier contains C_NCM_ADDR. When the DLM receives this message, the DLM increments the hop count in the frame, and forwards the frame through the other R-port. This message is discarded by the LNM or the device that generated the message.

5.3.3.4.5.6 NCM_LINE_START

NCMT: 0x05

This network control message is used to broadcast that the network topology has been automatically configured as a line network. This message is initiated by the DLM whose state is changed to LNM when the existing line network is divided into two line networks, or when a link failure is detected in a ring network and the network is reconfigured as a line network. This message is broadcast on the network using the broadcast address.

5.3.3.4.5.7 NCM_RING_START

NCMT: 0x06

This network control message is used to broadcast that the network topology has been automatically configured as a ring network. This message is initiated and broadcast through both R-ports by the DLM whose state is changed to RNMP.

5.3.3.4.5.8 NCM_ACK_RNMS

NCMT: 0x07

This network control message is used by the RNMS device to broadcast that the RNMS has been successfully selected. This message is transmitted from the RNMS to the RNMP.

5.3.3.4.5.9 NCM_RETRY_RNMS

NCMT: 0x08

This network control message is used to request a retransmission of the NCM_ACK_RNMS message from the RNMS device in the case where the RNMP device did not initially receive it. The NCM_RETRY_RNMS message is transmitted from the RNMP to the RNMS.

5.3.3.5 Extension (EXT)

5.3.3.5.1 General

This field exists when the VoE bit in the frame control field is set to TRUE. The extension field is specified as shown in Figure 10.

Extension Code															
G	Extension Type							Extension Length							
15	14	13	21	11	10	9	8	7	6	5	4	3	2	1	0

key
G: Group Mask

Figure 10 – Extension field

The separate field and its permissible values are described in Table 44.

Table 44 – Extension

Field Name	Position	Value/Description
Extension Length	Bit 0 – 7	0 to 255: the length of extension field
Extension Type	Bit 8 – 14	0: Invalid extension type 1 to 127: reserved for future use
Group Mask Enable	Bit 15	0x0: Group mask is enabled 0x1: Group mask is disabled

5.3.3.5.2 Group mask enable

Group Mask Enable is a bit field to specify whether the frame is to be accepted by the peer device or not, when the frame is broadcast or multicast. When the value is set to TRUE, group mask is enabled in the peer device that receives the frame. Otherwise, group mask is disabled in the peer device. When the group mask is enabled, the group mask fields are appended in the option field (see 5.3.3.8).

5.3.3.5.3 Extension type

This field indicates the type of extension field. The value 0x00 indicates an invalid extension type and the other values are reserved for future use.

5.3.3.5.4 Extension length

This field indicates the length of the extension field. When group mask enable is set to TRUE and extension type is set to 0x00, the extension length specifies the length of the group mask field. When group mask enable is set to FALSE and extension type is set to a value other than 0x00, extension length specifies the length of the extension field. When group mask enable is set to TRUE and extension type is not set to 0x00, the first two octets specify the length of the group mask field and the next two octets specify the extension type.

5.3.3.6 DSAP

This field indicates the SAP of the DLE to which the DLPDU is sent. The permissible values are in the range 0 to 65 535. The DSAP is specified as shown in Figure 11 and Table 45.

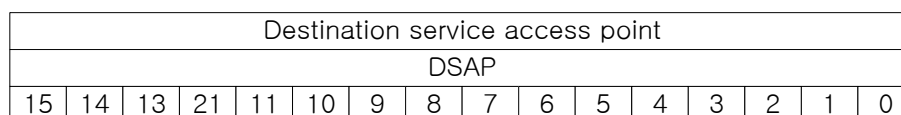


Figure 11 – DSAP field

Table 45 – Destination service access point

Field Name	Position	Value/Description
DSAP	Bit 0 – 15	Service access point of destination DLE

5.3.3.7 SSAP

This field indicates the SAP of the DLE from which the DLPDU is generated. The permissible values are in the range 0 to 65 535. The SSAP is specified as shown in Figure 12 and Table 46.

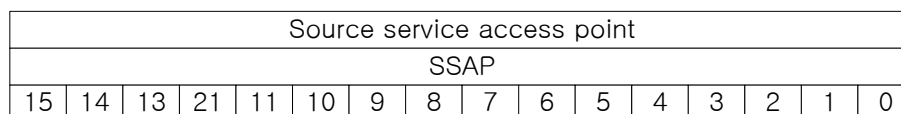


Figure 12 – Source service access point field

Table 46 – source service access point

Field Name	Position	Value/Description
SSAP	Bit 0 – 15	Service access point of source DLE

5.3.3.8 Option

5.3.3.8.1 General

This field indicates the option field when the VoE (see 5.3.3.4.2) is set to TRUE. The option field is used to indicate group mask information or other additional information. The maximum length of option field is limited to 256 octets.

5.3.3.8.2 Length of group mask and extension information

This field indicates the length of group mask and extension information. When group mask enable is set to TRUE and the extension type is not 0x00, the first two octets indicate the length of the group mask field and the next two octets indicate the length of the extension type field. When group mask enable is set to FALSE and the extension type is 0x00, the length of group mask and extension information field is ignored.

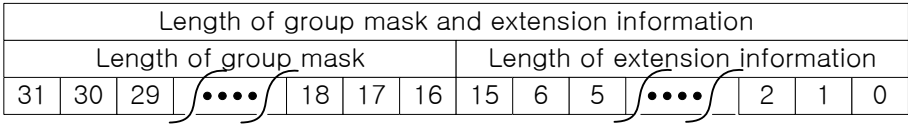


Figure 13 – Length of group mask and extension information

5.3.3.8.3 Group mask

This field uses a bit sequence to indicate the receipt selection of a message. When group mask enable is set to TRUE, the group mask field is appended in 4-octet units, for example, 4, 8, 12 ... 32 octets. Each bit indicates the receipt selection of the frame for the corresponding DL- entity identifier. A 1 means TRUE for frame receipt and 0 means FALSE. The first bit indicates the frame receipt option for highest DL- entity identifier. Figure 14 shows the bit sequence order of group mask field when its length is set to 255. When the extension type is set to 0x00, the group mask field is appended to the option field. Otherwise, the group mask field is appended after the length of group mask and extension information fields.

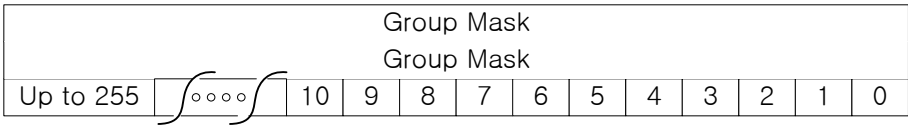


Figure 14 – Group mask option field

5.3.3.8.4 Extension information

This field is reserved for future extension. This field contains 4 octets of extension information.

5.3.3.9 Data and pad

This field indicates the data field received from a DLS-user.

5.4 Order of bit transmission

The order of bit transmission is identical to ISO/IEC 8802-3:2000, Clause 3. Each octet of the DLPDU with the exception the FCS is transmitted with the low-order bit first.

5.5 Invalid DLPDU

An invalid DLPDU shall be defined as one that meets at least one of the following conditions; this is almost identical to ISO/IEC 8802-3:2000, Clause 3:

- a) the frame length is inconsistent with a length value specified in the Length/type field. If the Length/type field contains a type value defined by ISO/IEC 8802-3:2000, 3.2.6, then the frame length is assumed consistent with this field and should not be considered an invalid DLPDU on this basis;
- b) it is not an integral number of octets in length;
- c) the bits of the incoming DLPDU excluding the FCS field do not generate a CRC value identical to the one received;

d) it is inconsistent with a F-type value of Type 21 fieldbus DLPDU.

The contents of invalid DLPDUs shall not be passed to the DL-user or DLE. The occurrence of an invalid DLPDU may be communicated to the network management.

NOTE Invalid DLPDUs may be ignored, discarded, or used in a private manner by a DL-user other than the RTE DL-user. The use of such DLPDUs is beyond the scope of this standard.

6 DLPDU structure and procedure

6.1 General

This clause defines the structure, contents, and encoding for each type and format of the DLPDU, along with procedural elements. Subclauses describe the structure, contents, parameters, and encoding of the DLPDU, along with the Type 21-specific part of the DLPDU structure, which is shown in Figure 4. The aspects relating to sending and receiving by DLS-users and their DLEs are also described.

NOTE In 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, and this is sometimes referred to as “little-endian” bit numbering.

6.2 Common DLPDU Field

6.2.1 General

Figure 15 shows the common DLPDU field. The Version and Length field is common to every DLPDU and is not described further for each DLPDU type.

Version and Length															
Version					Length										
15	14	13	21	11	10	9	8	7	6	5	4	3	2	1	0

Figure 15 – Common DLPDU field

6.2.2 Version

This field indicates the Type 21 protocol version with a 5 bit sequence. The highest 2 bits specify the major version and the lowest 3 bits specify the minor version (see 5.3.3.1).

6.2.3 Length

This field indicates the length of the data field in octets including the FCS field. The permissible values are in the range 12 to 1 498.

6.3 DL-DATA Transfer

6.3.1 DT DLPDU

6.3.1.1 General

The DT DLPDU is used to carry Type 21 data from one device to another.

6.3.1.2 DT DLPDU structure

Figure 16 shows the process of building a DT DLPDU and Figure 17 shows the structure of the DT DLPDU.

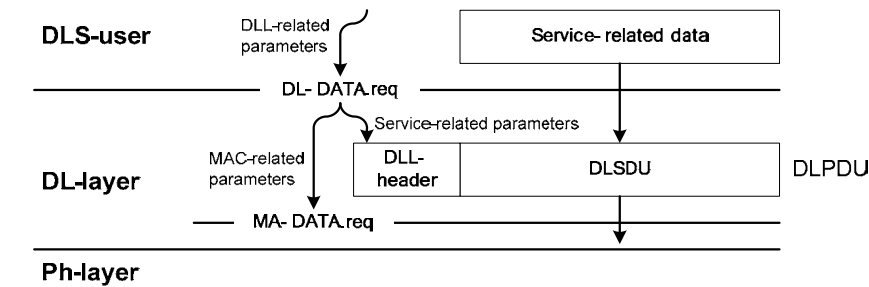


Figure 16 – Building a DT DLPDU

DST Addr	SRC Addr	FC	EXT	DSAP	SSAP	opt	DATA
----------	----------	----	-----	------	------	-----	------

Figure 17 – DT DLPDU structure

6.3.1.3 DT DLPDU Parameters

6.3.1.3.1 General

Table 47 shows the list of DT DLPDU parameters.

Table 47 – DT DLPDU parameters

Parameter	Data type	Value/Description
DST_addr	UNSIGNED16	Destination DL– entity identifier [Optional] -Broadcast -Multicast (user-defined) -Unicast
Priority	UNSIGNED8	0x00 to 0x03 0x04 to 0xFF: Reserved
DSAP	UNSIGNED16	Destination DLSAP of remote device
SSAP	UNSIGNED16	Source DLSAP
Group mask	—	Intended destination device bit array. Only when broadcast or multicast.
Group mask length	UNSIGNED8	Length of group mask. Only when broadcast or multicast.
Data	—	DLSDU
Data Length	UNSIGNED16	Length of DLSDU

6.3.1.3.2 DST_addr

This parameter indicates the destination DL– entity identifier of the DLE(s) for which the DLPDU is intended. This may be an individual or multicast (including broadcast) DL– entity identifier, not its MAC address. The destination address for unicast is explicitly assigned a value between 0x0000 and 0x00FF. A user-defined multicast address is assigned a value in the range 0xFFDE to 0xFFFD. A broadcast address is assigned the value 0xFFFF. The values 0x0100–0xFFDC are reserved. The value 0xFFDD indicates an invalid address (see 5.3.3.2).

6.3.1.3.3 Priority

This field indicates the frame priority with a 2 bit sequence. This field contains the value of the message priority parameter of the DL service. The highest priority is 0x03, and 0x00 is the lowest (see 5.3.3.4.3).

6.3.1.3.4 DSAP

See 5.3.3.6.

6.3.1.3.5 SSAP

See 5.3.3.7.

6.3.1.3.6 Group mask

See 5.3.3.8.3.

6.3.1.3.7 Group mask length

See 5.3.3.8.2.

6.3.1.3.8 Data

This data field is the DLSDU.

6.3.1.3.9 Data length

This field indicates the length of the data field in octets. The permissible values are in the range 0 to 1 486.

6.3.1.4 Sending

When the local DLS-user initiates a DL-DATA service request to transfer a DLSDU to a peer DLS-user, the DLSDU is stored in the RT-queue and transmitted by the DLPM using a DT DLPDU. The group mask option is available when the DL-DATA DLPDU is broadcast or multicast.

Table 48 shows the required data link service primitives and parameters to send a DT DLPDU. To send a DT DLPDU to a peer DLE, the DLPM queries the peer device's path information to the DLM using the DLM-GET_PATH service. Then, the DT DLPDU is transmitted using an MA-DATA service request primitive.

Table 48 – Primitives exchanged between DLS-user and DLE to send a DT DLPDU

Primitive	Source	Associated parameters
DL-DATA request	DLS-user	DST_addr DSAP SSAP Priority Group mask Group mask length Data Data length
DLM-GET_PATH request	DLE (DLPM)	DST_addr
DLM-GET_PATH confirm	DLM	Status R-port MAC address
MA-DATA request	DLE (DLPM)	DLPDU DLPDU length

6.3.1.5 Receiving

When a DT DLPDU is received through the MAC layer, the DLPM checks that it is valid and if it is intended for the local DLS-user. If the DT DLPDU is broadcast or multicast, the DLPM examines the group mask field to check if the DT DLPDU is intended for the local DLS-user. If the DLPDU is intended for the local DLS-user, the DLPM extracts the DLSDU from the received DT DLPDU and delivers the DLSDU to the appropriate DLS-user using a DL-DATA indication primitive. If no DLS-user is registered for the DSAP in the received DT DLPDU, the DLPM discards the DT DLPDU just received.

Table 49 shows the required data link service primitives and parameters to receive a DT DLPDU. When a DT DLPDU is received through the MA-DATA indication service primitive, the DLPM extracts the DLSDU from the received DT DLPDU and delivers the DLSDU to the appropriate DLS-user.

Table 49 – Primitives exchanged between DLS-user and DLEs to receive a DT DLPDU

Primitive	source	Associated parameters
MA-DATA indication	MAC	DLPDU DLPDU length
DL-DATA indication	DLPM	DLSDU DLSDU length

6.4 DL-SPDATA Transfer

6.4.1 SPDT DLPDU

6.4.1.1 General

SPDT DLPDUs are used to carry non-Type 21 data, such as IP.

6.4.1.2 SPDT DLPDU structure

Figure 18 shows the process of building an SPDT DLPDU and Table 50 shows the structure of the SPDT DLPDU. The Type 21 header is not included in the SPDT DLPDU.

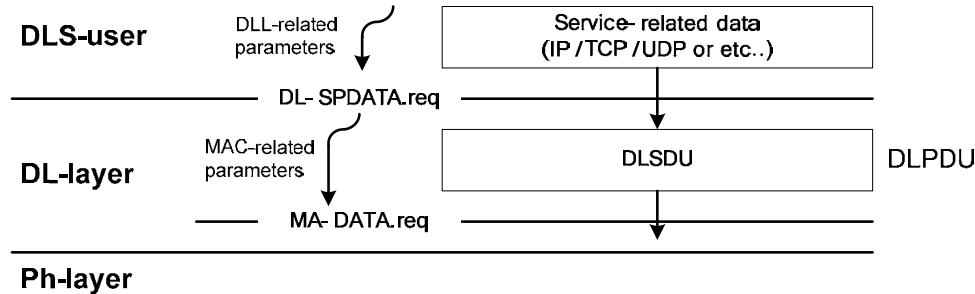


Figure 18 – SPDT DLPDU structure

6.4.1.3 SPDT DLPDU parameters

6.4.1.3.1 General

Table 50 shows the parameter list for the SPDT DLPDU.

Table 50 – SPDT DLPDU Parameters

Parameter	Data type	Value/Description
Data	Octetstring	DLSDU
Data Length	UNSIGNED16	Length of DLSDU

6.4.1.3.2 Data

This field indicates the DLSDU, which is to be transmitted in a standard ISO/IEC 8802-3:2000 Ethernet frame. This DLSDU can be any protocol data unit, such as an IP packet.

6.4.1.3.3 Data length

This field indicates the length of the data field in octets. The permissible values are in the range 0 to 1 514.

6.4.1.4 Sending

When the local DLS-user initiates a DL-SPDATA service request to transfer a DLSDU to the peer DLS-user, the DLSDU is stored in the NRT-queue and transmitted by the DLPM using an SPDT DLPDU. In this case, the DLSDU is transmitted as the SPDT DLPDU without a Type 21 header.

Table 51 shows the required data link service primitives and parameters to send an SPDT DLPDU. The SPDT DLPDU is transmitted by the DLPM using a MA-DATA request service primitive.

Table 51 – Primitive exchanged between DLS-User and DLEs to send an SPDT DLPDU

Primitive	Source	Associated parameters
DL-SPDATA request	DLS-User	Data: DLSDU Data length: length of DLSDU
MA-DATA request	DLPM	DLPDU DLPDU length
MA-DATA confirm	MAC	Status

6.4.1.5 Receiving

When an SPDT DLPDU is received through the MAC layer, the DLPM checks that it is valid and if it is intended for the local DLS-user. If this is the case, the DLPM extracts the DLSDU from the SPDT DLPDU and delivers it to the appropriate DLS-user using a DL-DATA indication primitive.

Table 52 shows the required data link service primitives and parameters to receive an SPDT DLPDU. When an SPDT DLPDU is received through the MA-DATA indication service primitive, the DLPM extracts the DLSDU from the received SPDT DLPDU and delivers the DLSDU to the appropriate local DLS-user.

Table 52 – Primitives exchanged between DLS-user and DLEs to receive an SPDT DLPDU

Primitive	Source	Associated parameters
MA-DATA indication	DLPM	DLPDU DLPDU length
DL-SPDATA indication	MAC	DLSDU DLSDU length

6.5 Network control messages

6.5.1 General

Network control message (NCM) DLPDUs are used to transfer the network control messages among DLMs. Five message types are provided to share the network information.

6.5.2 NCM_LA DLPDU

6.5.2.1 General

NCM_LA DLPDU is used to transfer local device information to the peer device when a link is established. When the DLM detects the change of link status through the Ph-LINK_STATUS_CHANGE indication primitive, the DLM queries the link status information of the physical layer. After that, the DLM generates an NCM_LA DLPDU to send the changed link information to the peer device over the newly established link.

6.5.2.2 NCM_LA DLPDU structure

NCM messages are based on the basic Type 21 DLPDU structure shown in Figure 19. Figure 19 shows the structure of the NCM_LA DLPDU after the Ethernet header.

DST Addr (C_NCM_ADDR)	SRC Addr	FC	EXT	DSAP	SSAP	DATA (Local Device Information)
--------------------------	----------	----	-----	------	------	------------------------------------

Figure 19 – NCM_LA DLPDU structure

6.5.2.3 NCM_LA DLPDU parameters

Table 53 shows the parameter list of the NCM_LA DLPDU.

Table 53 – NCM_LA DLPDU parameters

Parameter	Data type	Description
DST_addr	UNSIGNED16	C_NCM_ADDR (see 5.3.3.2)
SRC_addr	UNSIGNED16	Local DL– entity identifier DL_ADDR (see 5.3.3.3)
NCMT	UNSIGNED8	NCM_FAMILY_REQ (see 5.3.3.4.5.2)
DLMDU	—	DL management data unit Local device information (see 4.6.5)
Length	UNSIGNED16	Length of DLMDU
R-port	UNSIGNED8	Newly link-activated R-port

6.5.2.4 Sending

When the DLM detects the change in the link status through the Ph-LINK_STATUS_CHANGE indication primitive, the DLM queries the link status information of the physical layer. If the link is newly established, the DLM generates NCM_LA DLPDU to send the changed link information to the peer device over the newly established link. NCM_LA DLPDU is stored in the RT-queue and transmitted by the DLPM using the MA-DATA request primitive.

6.5.2.5 Receiving

The C_NCM_ADDR is used as the destination DL– entity identifier in the NCM_LA DLPDU. When an NCM_LA DLPDU is received through the MAC layer, the DLM increases the hop count field in the DLPDU and forwards the DLPDU through the other R-port. The NCM_LA DLPDU is discarded by the LNM or the device that generated the DLPDU. NCM_LA DLPDU is received and processed by the DLM as follows:

- the DLM checks whether the DLPDU is generated by the device itself. If this is the case, and the DLPDU is received through the other R-port, it means the network is configured as a ring network. If this is the case, proceed to step f) and finish the procedure without step g). Otherwise, proceed to step b);
- set the R-port's linked device type in the PORT_INFO (see 4.6.5.9) to “homogeneous device type”;
- extract the device path information (DPI) from the DLMDU and check for a DL– entity identifier collision using the DPI. If the source address is not registered in the DPI, generate an EVENT_IN_DEVICE event. If a local DL– entity identifier collision is detected, set the DEV_FLAG (see 4.6.5.3) to “Local DL– entity identifier Collision,” and generate an EVENT_THIS_ADDR_COLLISION event. If a network DL– entity identifier collision is detected, set the NET_FLAG (see 4.6.6.11) to “Network DL– entity identifier Collision Status,” and generate an EVENT_NET_ADDR_COLLISION event. Notification of any generated event is sent to the local DLMS-user using the DLM-EVENT indication service primitive;
- increment the HOP_CNT (see 4.6.5.13) in the received DLMDU;
- forward the modified NCM_LA DLPDU through the other R-port;

- f) check the DLM state trigger event condition using the PORT_INFO and perform the DLM state transition (see 7.3.3);
- g) generate an NCM_AT DLPDU and transmit the DLPDU through the opposite R-port.

6.5.3 NCM_AT DLPDU

6.5.3.1 General

An NCM_AT DLPDU is used to transfer local device information to the other devices on the network when a device receives an NCM_LA DLPDU. This DLPDU is transmitted through the R-port that is used to receive the NCM_LA DLPDU.

6.5.3.2 NCM_AT DLPDU structure

See 6.5.2.2.

6.5.3.3 NCM_AT DLPDU parameters

Table 54 shows the parameter list of NCM_AT DLPDU.

Table 54 – NCM_AT DLPDU parameters

Parameter	Data type	Description
DST_addr	UNSIGNED16	C_NCM_ADDR (see 5.3.3.2.3)
SRC_addr	UNSIGNED16	Local DL– entity identifier DL_ADDR (see 4.6.5.2)
NCMT	UNSIGNED8	NCM_ADV_THIS (see IEC 61158-3-21:2010, 4.4.3.2.4)
DLMDU	—	Local device information (see 4.6.5)
Length	UNSIGNED16	Length of DLMDU
R-port	UNSIGNED8	R-port used to receive the NCM_LA DLPDU

6.5.3.4 Sending

When the DLM receives an NCM_LA DLPDU, the DLM generates an NCM_AT DLPDU for the DLMDU including the local device information. The DLMDU is stored in the RT-queue and transmitted by the DLMP through the R-port through which the NCM_LA DLPDU was received. In this case, C_NCM_ADDR is used as the destination DL– entity identifier.

6.5.3.5 Receiving

The C_NCM_ADDR is used as the destination address in an NCM_AT DLPDU. When an NCM_AT DLPDU is received through the MAC layer, the DLM increments the hop count field in the DLPDU and forwards the DLPDU through the other R-port. The NCM_AT DLPDU is discarded by the LNM or the device that generated the DLPDU. The NCM_AT DLPDU is received and processed by the DLM as follows:

- a) the DLM checks whether the DLPDU was generated by the device itself. If the DLPDU was generated locally and received through the other R-port, it means the network is configured as a ring network. In this case, proceed to step f). Otherwise, proceed to step b);
- b) set the R-port's linked device type in the PORT_INFO (see 4.6.5.9) to “homogeneous device type”;
- c) extract the device path information (DPI) (see 4.6.7) from the DLMDU and check for a DL– entity identifier collision using the DPI. If the source address is not registered in the DPI, generate an EVENT_IN_DEVICE event. If a local DL– entity identifier collision is detected, set the DEV_FLAG (see 4.6.5.3) to “Local DL– entity identifier Collision,” and generate an EVENT_THIS_ADDR_COLLISION event. If a network DL– entity identifier collision is

detected, set the NET_FLAG (see 4.6.6.11) to “Network DL– entity identifier Collision Status,” and generate an EVENT_NET_ADDR_COLLISION event. Notification of any generated event is sent to the local DLMS-user using the DLM-EVENT indication service primitive;

- d) increment HOP_CNT (see 4.6.5.13) in the received DLMDU;
- e) forward the modified NCM_LA DLPDU through the other R-port;
- f) check the DLM state trigger event condition using the PORT_INFO and perform the DLM state transition (see 7.3.3).

6.5.4 NCM_LS DLPDU

6.5.4.1 General

The NCM_LS DLPDU is used to indicate that the network is automatically configured as a line topology when a line network is newly established, or that an existing line network is divided into two lines, or that a ring network has been reconfigured as a line network. This DLPDU is generated by the device that is selected as the LNM on the network.

6.5.4.2 NCM_LS DLPDU structure

See 6.5.2.2.

6.5.4.3 NCM_LS DLPDU parameters

Table 55 shows the NCM_LS DLPDU parameter list.

Table 55 – NCM_LS DLPDU parameters

Parameter	Data type	Description
DST_addr	UNSIGNED16	Broadcast address(see 5.3.3.2.2)
SRC_addr	UNSIGNED16	Local DL– entity identifier DL_ADDR (see 4.6.5.2)
NCMT	UNSIGNED8	NCM_LINE_START (see 5.3.3.4.5.4)
DLMDU	—	Local device information (see 4.6.5)
Length	UNSIGNED16	Length of DLMDU
R-port	UNSIGNED8	link-activated R-port

6.5.4.4 Sending

When the state of the DLM is changed to LNM, the DLM generates an NCM_LS DLPDU to inform every device on the network of the change in network topology. The NCM_LS DLPDU is stored in the RT-queue and it is broadcast by the DLPM using the MA-DATA request primitive.

6.5.4.5 Receiving

The broadcast DL– entity identifier is used as the destination address in an NCM_LS DLPDU. Therefore, the NCM_LS DLPDU is directly forwarded by the MAC layer. The NCM_LS DLPDU is processed by the DLM as follows:

- a) extract the DPI (see 4.6.7) from the DLMDU and use it to check for a DL– entity identifier collision. If the source address is not registered in the DPI, generate an EVENT_IN_DEVICE event. If a local DL– entity identifier collision is detected, set the DEV_FLAG (see 4.6.5.3) to “Local DL– entity identifier Collision,” and generate an EVENT_THIS_ADDR_COLLISION event. If a network DL– entity identifier collision is detected, set the NET_FLAG (see 4.6.6.11) to “Network DL– entity identifier Collision

Status,” and generate an EVENT_NET_ADDR_COLLISION event. Notification of any generated event is sent to the local DLMS-user using the DLM-EVENT indication service primitive;

- b) check the DLM state trigger event condition using the PORT_INFO and perform the DLM state transition (see 7.3.3);
- c) if the DLM state is not LNM or RNM, enable the frame forward functions and set “frame forwarding function from R-port1 to R-port2” and “Frame Forwarding Function from R-port2 to R-port1” in the PORT_INFO (see 4.6.5.9) to TRUE.

6.5.5 NCM_RS DLPDU

6.5.5.1 General

The NCM_RS DLPDU is used to indicate that the network is automatically configured as a ring topology. This DLPDU is generated by the device that is selected as the RNMP on the network.

6.5.5.2 NCM_RS DLPDU structure

See 6.5.2.2.

6.5.5.3 NCM_RS DLPDU parameters

Table 56 shows the NCM_RS DLPDU parameter list.

Table 56 – NCM_RS DLPDU parameters

Parameter	Data type	Description
DST_addr	UNSIGNED16	Broadcast address
SRC_addr	UNSIGNED16	Local DL– entity identifier, DL_ADDR (see 4.6.5.2)
NCMT	UNSIGNED8	NCM_RING_START (see IEC 61158-3-21:2010 4.4.3.2.4)
DLMDU	—	Local device information (see 4.6.5)
Length	UNSIGNED16	Length of DLMDU
R-port	UNSIGNED8	R-port1, R-port2

6.5.5.4 Sending

When the state of the DLM is changed to RNMP, the DLM generates an NCM_RS DLPDU to inform every device on the network of the change in network topology. The NCM_RS DLPDU is stored in the RT-queue and broadcast by the DLPM using the MA-DATA request primitive.

6.5.5.5 Receiving

The broadcast DL– entity identifier is used as the destination address in an NCM_RS DLPDU. Therefore, the NCM_RS DLPDU is directly forwarded by the MAC layer. The NCM_RS DLPDU is processed by the DLM as follows:

- a) extract the DPI (see 4.6.7) from the DLMDU and use it to check for a DL– entity identifier collision. If the source address is not registered in the DPI, generate an EVENT_IN_DEVICE event. If a local DL– entity identifier collision is detected, set the DEV_FLAG (see 4.6.5.3) to “Local DL– entity identifier Collision,” and generate an EVENT_THIS_ADDR_COLLISION event. If a network DL– entity identifier collision is detected, set the NET_FLAG (see 4.6.6.11) to “Network DL– entity identifier Collision Status,” and generate an EVENT_NET_ADDR_COLLISION event. Notification of any

generated event is sent to the local DLMS-user using the DLM-EVENT indication service primitive;

- b) check the DLM state trigger event condition using the PORT_INFO and perform the DLM state transition (see 7.3.3);
- c) if the DLM state is not RNMP or RNMS, enable the frame forward functions and set “frame forwarding function from R-port1 to R-port2” and “Frame Forwarding Function from R-port2 to R-port1” in PORT_INFO (see 4.6.5.9) to TRUE.
- d) when the device is designated as the RNMS by the RNMP, disable the frame forwarding function in the RNMP direction;
- e) enter the RNMS state if the frame forwarding control is successfully completed. Otherwise, remain in the GD state;
- f) when the DLM enters the RNMS state, send NCM_ACK_RNMS to the RNMP.

6.5.6 NCM_AR_DLPDU

6.5.6.1 General

The NCM_AR DLPDU is used to indicate that the designated RNMS has been successfully assigned and is operational. This DLPDU is generated by the device that is selected as the RNMS and transferred to the RNMP.

6.5.7 NCM_AR DLPDU structure

6.5.7.1 General

See 6.5.2.2.

6.5.7.2 NCM_AR DLPDU parameters

Table 57 shows the NCM_AR DLPDU parameter list.

Table 57 – NCM_AR DLPDU parameters

Parameter	Data type	Description
DST_addr	UNSIGNED16	DL– entity identifier of RNMP
SRC_addr	UNSIGNED16	Local DL– entity identifier DL_ADDR (see 4.6.5.2)
NCMT	UNSIGNED8	NCM_ACK_RNMS (see IEC 61158-3-21:2010, 4.4.3.2.4)
DLMDU	—	Local device information (see 4.6.5)
Length	UNSIGNED16	Length of DLMDU
R-port	UNSIGNED8	Destination R-port in the Path table

6.5.7.3 Sending

When the state of the DLM changes from GD to RNMS, the DLM generates an NCM_AR DLPDU to indicate that the designated RNMS is successfully assigned, and unicasts it to the RNMP.

6.5.7.4 Receiving

The NCM_AR DLPDU is unicast from the RNMS to the RNMP. The NCM_RS DLPDU is processed by the DLM in the RNMP device as follows:

- a) extract the DPI (see 4.6.7) from the DLMDU and use it to check for a DL– entity identifier collision. If the source address is not registered in the DPI, generate an

EVENT_IN_DEVICE event. If a local DL– entity identifier collision is detected, set the DEV_FLAG (see 4.6.5.3) to “Local DL– entity identifier Collision,” and generate an EVENT_THIS_ADDR_COLLISION event. If a network DL– entity identifier collision is detected, set the NET_FLAG (see 4.6.6.11) to “Network DL– entity identifier Collision Status,” and generate an EVENT_NET_ADDR_COLLISION event. Notification of any generated event is sent to the local DLMS-user using the DLM-EVENT indication service primitive;

- b) Disable the frame forwarding function in the RNMS direction.

7 DLE elements of procedure

7.1 Overall structure

The DLL is composed of the control elements of the DLPM, the dual MAC (DMAC), the dual physical interface (DPHY), and the DLL management Interface. The DLPM is the primary control element. It provides the functions for deterministic MAC by coordinating the DMAC and the NCMs for reliable and efficient support both of higher-level connectionless real-time and non-real-time data transfer services. The DLL management interface provides DLL management functions. Figure 20 depicts the overall structure of the DLL.

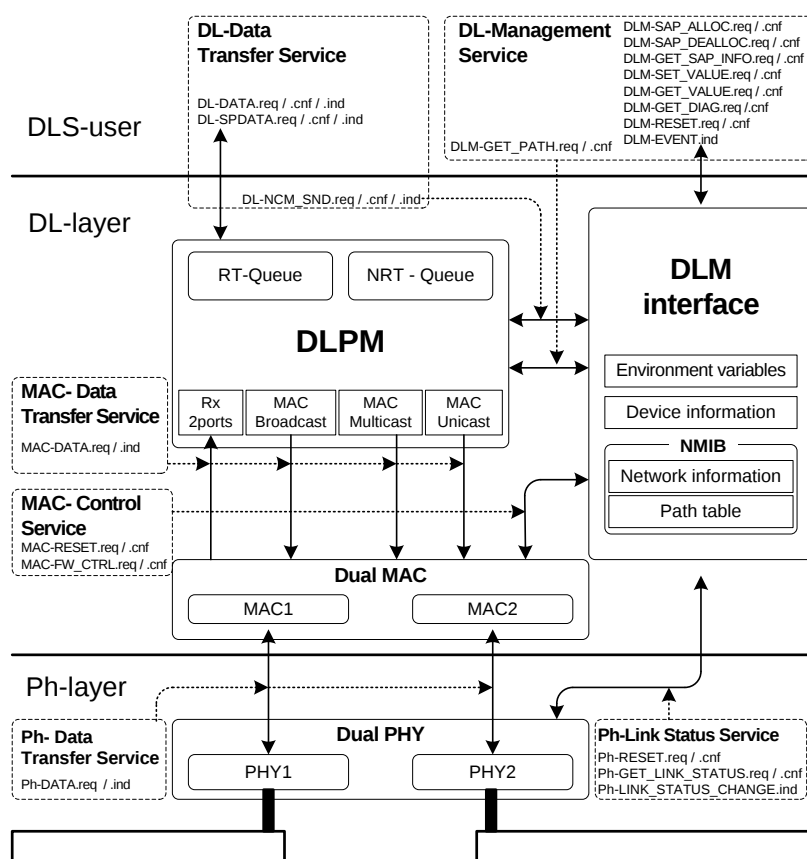


Figure 20 – DLL structure and elements

7.2 DL-protocol machine (DLPM)

7.2.1 Overview

The DLPM maintains two transmitter queues and one receiver queue. When a local DLS-user or DLMS-user generates a message, the message is stored in the RT-queue or the NRT-queue in the form of a DLPDU. The DLPM handles the messages in the RT-queue according to the DLPMS. The DLPM transmits the message using the MA-DATA request

primitive. When a message is received by the MAC layer, the received message is stored in the receiver queue and processed on a first-in, first-out basis by the DLPM.

7.2.2 Primitive definitions

7.2.2.1 Primitives exchanged between DLPM and DLS-user

Table 58 shows the data link service primitives exchanged between the DLPM and the DLS-user.

Table 58 – Primitives exchanged between DLPM and DLS-user

Primitive	Source	Associated parameters	Description
DL-DATA.req	DLS-user	DST_addr DSAP SSAP Priority DLSDU DLSDU length	Transmit request to remote Type 21 device
DL-DATA.cnf	DLPM	Status	Confirmation to calling DLS-user
DL-DATA.ind	DLPM	DST_addr SRC_addr SSAP DLSDU DLSDU length	Receive indication from a remote Type 21 device
DL-SPDATA.req	DLS-user	DLSDU DLSDU length	Sporadic data transmit request to remote Type 21 device
DL-SPDATA.cnf	DLPM	Status	Confirmation to calling DLS-user
DL-SPDATA.ind	DLPM	DLSDU DLSDU length	Sporadic data receive indication from a remote Type 21 device

Table 59 shows the parameters exchanged between the DLPM and the DLS-user.

Table 59 – Parameters exchanged between DLPM and DLS-user

Parameter	Description
DST_addr	Destination DL– entity identifier. C_BROADCAST_ADDR: broadcast address 0xFFFF User-defined multicast address: 0xFFFFD–0xFFDE 0-MAX_DEVICE_ADDR: unicast address
DSAP	Destination service access point
SSAP	Source service access point
Priority	Message priority. Permissible values are C_PRI_0 through C_PRI_3 C_PRI_3 is the highest priority.
DLSDU	DLSDU
DLSDU length	Length of DLSDU
Status	This parameter allows the DLMS-user to determine whether the requested DLMS was provided successfully. If it failed, the reason is specified. The value of this parameter is one of: “OK – success – the variable could be updated”; “Failure – the variable does not exist or could not assume the new value”; “Failure – invalid parameters in the request”.

NOTE C_NCM_ADDR is not used for DL-DATA and DL-SPDATA services.

7.2.2.2 Primitives exchanged between DLPM and DLM

Table 60 and Table 61 shows the data link service primitives and parameters exchanged between the DLPM and the DLM.

Table 60 – Primitives exchanged between DLPM and DLM

Primitive	Source	Associated parameters	Description
DLM-GET_PATH.req	DLPM	DST_addr	Path information request to DLM
DLM-GET_PATH.cnf	DLM	Status R-port MAC address	Confirmation to calling DLPM with R-port and MAC address
DLM-NCM_SND.req	DLM	DST_addr NCMT DLMDU Length R-port	Transmit request for network control message NCM_FAMILY_REQ NCM_FAMILY_RES NCM_MEDIA_LINKED NCM_ADV_THIS NCM_LINE_START NCM_RING_START NCM_ACK_RNMS NCM_RETRY_RNMS
DLM-NCM_SND.cnf	DLPM	Status	Confirmation to calling service user
DLM-NCM_SND.ind	DLPM	DST_addr SRC_addr NCMT DLMDU length R-port	Network control message receive indication from a remote Type 21 device

Table 61 – Parameters used with primitives exchanged between DLPM and DLM

Parameter	Description
DST_addr	Destination DL– entity identifier. C_BROADCAST_ADDR: broadcast address, 0xFFFF C_NCM_ADDR: Fixed DL– entity identifier for network control message, 0xFFFE 0-MAX_DEVICE_ADDR: unicast address
SRC_addr	Source DL– entity identifier. “0-MAX_DEVICE_ADDR: unicast address”
R-port	R-port which is used to transmit a frame or to receive a frame.
MAC address	ISO/IEC 8802-3:2000 MAC address that corresponds to the Destination DL– entity identifier.
NCMT	Network Control Message Type. NCM_FAMILY_REQ (0x01), NCM_FAMILY_RES (0x02), NCM_MEDIA_LINKED (0x03), NCM_ADV_THIS (0x04), NCM_LINE_START (0x05), NCM_RING_START (0x06), NCM_ACK_RNMS (0x07), NCM_RETRY_RNMS (0x08)
DLMDU	DL management data unit. The data field of DLM network control message.
Length	Length of DLMDU
Status	This parameter allows the DLMS-user to determine whether or not the requested DLMS was provided successfully. If it failed, the reason is specified. The value of this parameter can be one of: “OK – success – the variable could be updated”; “Failure – the variable does not exist or could not assume the new value”; “Failure – invalid parameters in the request”.

NOTE User-defined multicast addresses are not used for the destination DL– entity identifier.

7.2.3 DLPM state table

Figure 21 depicts the state transition diagram of the DLPM. The state table of the DLPM is shown in Table 62.

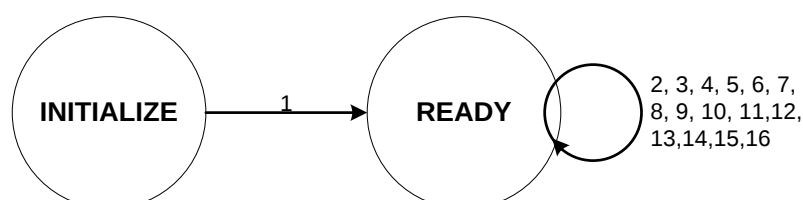
**Figure 21 – State transition diagram of the DLPM**

Table 62 – DLPM state table

#	Current	Event /Condition =>actions	Next state
1	INITIALIZE	POWER-ON or RESET / =>	READY
2	READY	DL-DATA.req {DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = "True" && DLM-GET_PATH.req {DST_addr} Status := DLM-GET_PATH.cfm { D_MAC_addr, R-port } Status = "Success" && CHECK_RTQUEUE() <> "Full" => DLPDU := BUILD_DLPDU(D_MAC_addr, DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length) ENQUEUE_RT(Priority, DLPDU, DLPDU length, R-port) DL-DATA.cfm{Status := "Success"}	READY
3	READY	DL-DATA.req {DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) <> " True" => DL-DATA.cfm{Status := "Failure – Invalid parameter"}	READY
4	READY	DL-DATA.req {DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = " True" && DLM-GET_PATH.ind { D_MAC_addr, R-port} ="Failure" => DL-DATA.cfm{Status := "Failure – not available destination"}	READY
5	READY	DL-DATA.req { DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = " True" && DLM-GET_PATH.ind { D_MAC_addr, R-port } =" Success" && CHECK_RTQUEUE() = "Full" => DL-DATA.cfm{Status := "Failure – The RT-queue is full"}	READY

#	Current	Event /Condition =>actions	Next state
6	READY	DL-SPDATA.req {DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = "True" && CHECK_NRTQUEUE() <> "Full" => DLPDU := DLSDU R-port := R-port1 && R-port2 ENQUEUE_NRT(DLPDU, DLPDU length, R-port) DL-SPDATA.cfm {Status := "Success"}	READY
7	READY	DL-SPDATA.req {DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = "False" => DL- SPDATA.cfm{Status := "Failure – Invalid requested parameter"}	READY
8	READY	DL-SPDATA.req {DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = "True" && CHECK_NRTQUEUE() = "Full" => DL- SPDATA.cfm{Status := "Failure – The NRT-queue is full"}	READY
9	READY	DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} / CHECK_PARA(Length) = "True" && CHECK_RTQUEUE() <> "Full" => Priority := C_HIGHEST_PRIORITY DLPDU := BUILD_DLPDU_NCM(DST_addr , Priority, NCMT , DLMDU, Length) QUEUE_RT(Priority, DLPDU, DLPDU length, R-port) DL-NCM_SND.cfm{Status := "Success"}	READY
10	READY	DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} / CHECK_PARA(Length) = "False" => DL- NCM_SND.cfm{Status := "Failure – Invalid parameter"}	READY
11	READY	DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} / CHECK_PARA(Length) = "True" && CHECK_RTQUEUE() = "Full" => DL- NCM_SND.cfm{Status := "Failure – The RT-queue is full"}	READY

#	Current	Event /Condition =>actions	Next state
12	READY	MA-DATA.ind{DLPDU, DLPDU length, R-port} / CHECK_FRAME_ETH_TYPE(DLPDU) = "Type 21" && CHECK_FRAME_FOR_THIS(DLPDU) = "True" && CHECK_FC(DLPDU) <> "Network control message" => DST_addr := GET_D_ADDR(DLPDU) SRC_addr := GET_S_ADDR(DLPDU) DSAP := GET_DSAP(DLPDU) SSAP := GET_SSAP(DLPDU) DLSDU := GET_DLSDU(DLPDU) DLSDU length := GET_DLSDU(DLPDU length) DLS-user := GET_DLS_USER(DSAP) dst_dls_user := SET_DLS_USER(DLS-user) dst_dls_user.DL-DATA.ind{ DST_addr, SRC_addr, SSAP, DLSDU, DLSDU length}	READY
13	READY	MA-DATA.ind{DLPDU, DLPDU length, R-port} / CHECK_FRAME_ETH_TYPE(DLPDU) <> "Type 21" => DLSDU := DLPDU DLSDU length := DLPDU length DL-SPDATA.ind{DLSDU, DLSDU length}	READY
14	READY	MA-DATA.ind{DLPDU, DLPDU length, R-port} / CHECK_FRAME_ETH_TYPE(DLPDU) = "Type 21" && CHECK_FRAME_FOR_THIS(DLPDU) = "True" && CHECK_FC(DLPDU) = "Network Control Message" => DST_addr := GET_D_ADDR(DLPDU) SRC_addr := GET_S_ADDR(DLPDU) DLMDU := GET_DLSDU(DLPDU) length := GET_DLSDU(DLPDU length) cmd := GET_NCM_CMD(DLPDU) DL-NCM_SND.ind{DST_addr, SRC_addr, NCMT, DLMDU, DLMDU length, R-port}	READY
15	READY	New DLPDU from DLS-user / CHECK_RTQUEUE() <> "Empty" => DLPDU := DQUEUE_RT(policy) MA-DATA.req {DLPDU, DLPDU length, R-port}	READY

#	Current	Event /Condition =>actions	Next state
16	READY	New DLPDU from DLS-user / CHECK_RTQUEUE() = "Empty" && / CHECK_NRTQUEUE() <> "Empty" => DLPDU := DQUEUE_NRT() MA-DATA.req {DLPDU, DLPDU length, R-port}	READY

7.2.4 DLPM functions

All functions of the DLPM are summarized in Table 63.

Table 63 – DLPM functions table

Function name	Input	Output	Operation
CHECK_PARA	DLSDU length	True/False	Return FALSE if the DLSDU exceeds the C_MAX_USER_DLMDU_SIZE. Otherwise, return TRUE.
CHECK_RTQUEUE	(none)	status	Check the RT-queue condition. The returned status is "Full," "Empty" or "Queued."
BUILD_DLPDU	D_MAC_addr DST_addr DSAP SSAP Priority DLSDU DLSDU length	DLPDU DLPDU length	Build DLPDU and return the DLPDU and its length.
ENQUEUE_RT	Priority DLPDU DLPDU length R-port	(none)	Queues the input data into the tail of the RT-queue.
CHECK_NRTQUEUE	(none)	status	Check that the NRT-Queue condition for DATA is fully queued. The returned status is "Full," "Empty" or "Queued."
ENQUEUE_NRT	DLPDU DLPDU length R-port	(none)	Queues the input data into the NRT-QUEUE on a FIFO basis.
BUILD_DLPDU_NCM	DST_addr Priority NCMT DLMDU DLMDU length	DLPDU DLPDU length	Build NCM DLPDU and return the DLPDU and its length.
DQUEUE_RT	Policy	DLPDU DLPDU length R-port	Get a DLPDU from the RT-queue as specified by the DLPMSF.
DQUEUE_NRT	(none)	DLPDU DLPDU length R-port	Get a DLPDU from the NRT-queue according to the FIFO method.
CHECK_FRAME_ETH_TYPE	DLPDU	status	Check Type 21 EtherType (0x88FE). Return "Type 21" when the type is correct. Otherwise, return "Other."
CHECK_FRAME_FOR_THIS	DLPDU	True/False	Check if the frame has been unicast to local device. Return TRUE if the Destination DL- entity identifier is equal to the Local DL- entity identifier. Otherwise, return FALSE.

Function name	Input	Output	Operation
CHECK_FC	DLPDU	status	Check if the received frame is a network control message. Return "Network control message" if the FC field indicates NCM. Otherwise, return "Other."
GET_D_ADDR	DLPDU	DST_addr	Return the Destination DL- entity identifier of the DLPDU.
GET_S_ADDR	DLPDU	SRC_addr	Return the Source DL- entity identifier of the DLPDU.
GET_DSAP	DLPDU	DSAP	Return the Destination SAP of the DLPDU.
GET_SSAP	DLPDU	SSAP	Return the Source SAP of the DLPDU.
GET_DLSDU	DLPDU	DLSDU	Return DLSDU of the DLPDU.
GET_DLSDU LENGTH	DLPDU length	DLSDU length	Return the DLSDU length.
GET_DLS_USER	DSAP	DLS-user ID	Return the DLS-user ID that owns the DSAP.
SET_DLS_USER	DLS-user ID	DLS-user object	Return the DLS-user object using the DLS-user ID.
GET_NCMT	DLPDU	NCMT	Return the NCMT of the NCM DLPDU.

7.3 DLL management Protocol

7.3.1 Overview

This clause describes the interface protocol between the DLM and DLMS-user. This description of the DLL management protocol provides the DLL management services specified in Clause 7 by making use of the services available to the DLMS-user. Full implementation details and matters of local responsibility are not included in this clause.

7.3.2 Primitive definitions

7.3.2.1 Primitive exchanged between DLM and DLS-user

Table 64 summarizes all primitives exchanged between the DLM and the DLS-user.

Table 64 – Primitives exchanged between DLM and DLS-user

Primitive	source	Associated parameters	Description
DLM-RESET.req	DLS-user	<none>	This request primitive causes the DLM to reset the DLE.
DLM-RESET.cnf	DLM	DLM-status	This indicates the status of the reset.
DLM-SET_VALUE.req	DLS-user	Variable name Desired value	This service is used to assign new values to the DLE variables.
DLM-SET_VALUE.cnf	DLM	Status	The DLMS-user receives confirmation that the specified variables have been set to the new values.
DLM-GET_VALUE.req	DLS-user	Variable name	This service is used to read the value of a DLE variable.
DLM-GET_VALUE.cnf	DLM	Status Current value	This service returns the actual value of the specified variable.
DLM-SAP_ALLOC.req	DLS-user	SAP DLS-user ID	This service is used by the DLMS-user to obtain a SAP assignment from the DLM.
DLM-SAP_ALLOC.cnf	DLM	Status	This service returns the result status of DLM-SAP_ALLOC.req.
DLM-SAP_DEALLOC.req	DLS-user	SAP	This service is used by the DLMS-user to release and return the allocated SAP to the DLM.
DLM-SAP_DEALLOC.cnf	DLM	Status	This service returns the result status of DLM-SAP_DEALLOC.req.
DLM-GET_SAP_INFO.req	DLS-user	SAP	This service is used by the DLMS-user to obtain the information of already allocated SAP from the DLM.
DLM-GET_SAP_INFO.cnf	DLM	Status DLS-user ID	This service returns the result status of DLM-GET_SAP_INFO.req, specifically the result status and DLS-user ID.
DLM-GET_DIAG.req	DLS-user	Diag-type Addr	This service is used by the DLMS-user to obtain the diagnostic information from the DLM.
DLM-GET_DIAG.cnf	DLM	Status Diag	This service returns the result status of DLM-GET_SAP_INFO.req, specifically the result status and diagnostic information.
DLM-EVENT.ind	DLM	Event	This service is used to inform the DLMS-user about certain events or errors in the DLL.

The parameters used with the primitives exchanged between the DLM and the DLS-user are described in Table 65.

Table 65 – Parameters used with primitives exchanged between DLM and DLS-user

Parameter	Description
DLM-status	This parameter allows the DLMS-user to determine whether or not the requested DLMS was provided successfully. If it failed, the reason is specified. The value of this parameter can be one of: “OK – successfully completed” “Failure – terminated before completion”
Variable name	This parameter specifies the DLE variable whose value is to be set.
Desired value	This parameter specifies the desired value for the selected variable.
Status	This parameter allows the DLMS-user to determine whether or not the requested DLMS was provided successfully. If it failed, the reason is specified. The value of this parameter can be one of: “OK – success – the variable could be updated” “Failure – the variable does not exist or could not assume the new value” “Failure – invalid parameters in the request”
Current value	This parameter indicates the current value of the designated variable.
DLS-user ID	This parameter indicates the numeric identification of the local DLS-user. DLS-user ID is unique in a device.
SAP	This parameter indicates the DLSAP.
Diag-type	This parameter indicates the type of diagnostic information. “DIAG_TYPE_L_DEVICE_INFO – local device information,”; “DIAG_TYPE_R_DEVICE_INFO – remote device information,”; “DIAG_TYPE_NET_INFO – network information”.
Addr	This parameter indicates the DL- entity identifier of the designated node.
Event	This parameter specifies the primitive or composite event being announced. The possible values are defined in the corresponding part of Clause 8.2.

7.3.2.2 Primitives exchanged between DLM and DLPM

Table 60 summarizes all primitives exchanged between the DLM and the DLPM.

7.3.2.3 Primitives exchanged between DLM and DMAC

Table 66 and Table 67 summarize all primitives and parameters exchanged between the DLM and the DMAC.

Table 66 – Primitive exchanged between DLM and DMAC

Primitive name	Source	Associated parameters	Description
MAC-RESET.req	DLM	<none>	Reset MAC layer.
MAC-FW_CTRL.req	DLM	R-port F_en	This service is used to control the frame forward functions in the MAC layer.
MAC-FW_CTRL.cnf	D-MAC	Status	This service returns the result status of MAC-FW_CTRL.req.

Table 67 – Parameters used with primitives exchanged between DLM and DMAC

Parameter name	Description
R-port	This parameter indicates the Type 21 Ethernet port. Every Type 21 device has two R-ports: R-port1 and R-port2.
F_en	This parameter indicates the frame forward control status between the two R-ports: “Enable - enable the hardware-based frame forward function” “Disable - disable the hardware-based frame forward function”
Status	This parameter allows the DLMS-user to determine whether or not the requested DLMS was provided successfully. If it failed, the reason is specified. The value of this parameter can be one of: “OK – success – the variable could be updated”; “Failure – the variable does not exist or could not assume the new value”; “Failure – invalid parameters in the request”.

7.3.2.4 Primitive exchanged between DLM and DPHY

Table 68 and Table 69 summarize all primitives and parameters exchanged between the DLM and the DPHY.

Table 68 – Primitive exchanged between DLM and DPHY

Primitive name	Source	Associated Parameters	Description
Ph-RESET.req	DLM	<none>	Reset physical layer.
Ph-GET_LINK_STATUS.req	DLM	R-port	This service is used to obtain link status information from the physical layer.
Ph-GET_LINK_STATUS.cnf	DPHY	R-port L_status	This service returns the result status of Ph-GET_LINK_STATUS.req
Ph-LINK_STATUS_CHANGE.ind	DPHY	R-port	This service is used to notify the DLM of the link status change event.

Table 69 – Parameters used with primitives exchanged between DLM and DPHY

Parameter name	Description
R-port	This parameter indicates the Type 21 Ethernet port. Every Type 21 device has two R-ports: R-port1 and R-port2.
L_status	“link active – The link is activated and available for communication” “link inactive – The link is not activated and not available for communication”

7.3.3 DLM state table

The DLM is maintained with 6 states: INITIALIZE, SA, LNM, GD, RNMP, and RNMS. The DLM controls the frame forward functions in the MAC layer according to its DLM state. DLM state transition diagram and descriptions are shown in Figure 22 and Table 70.

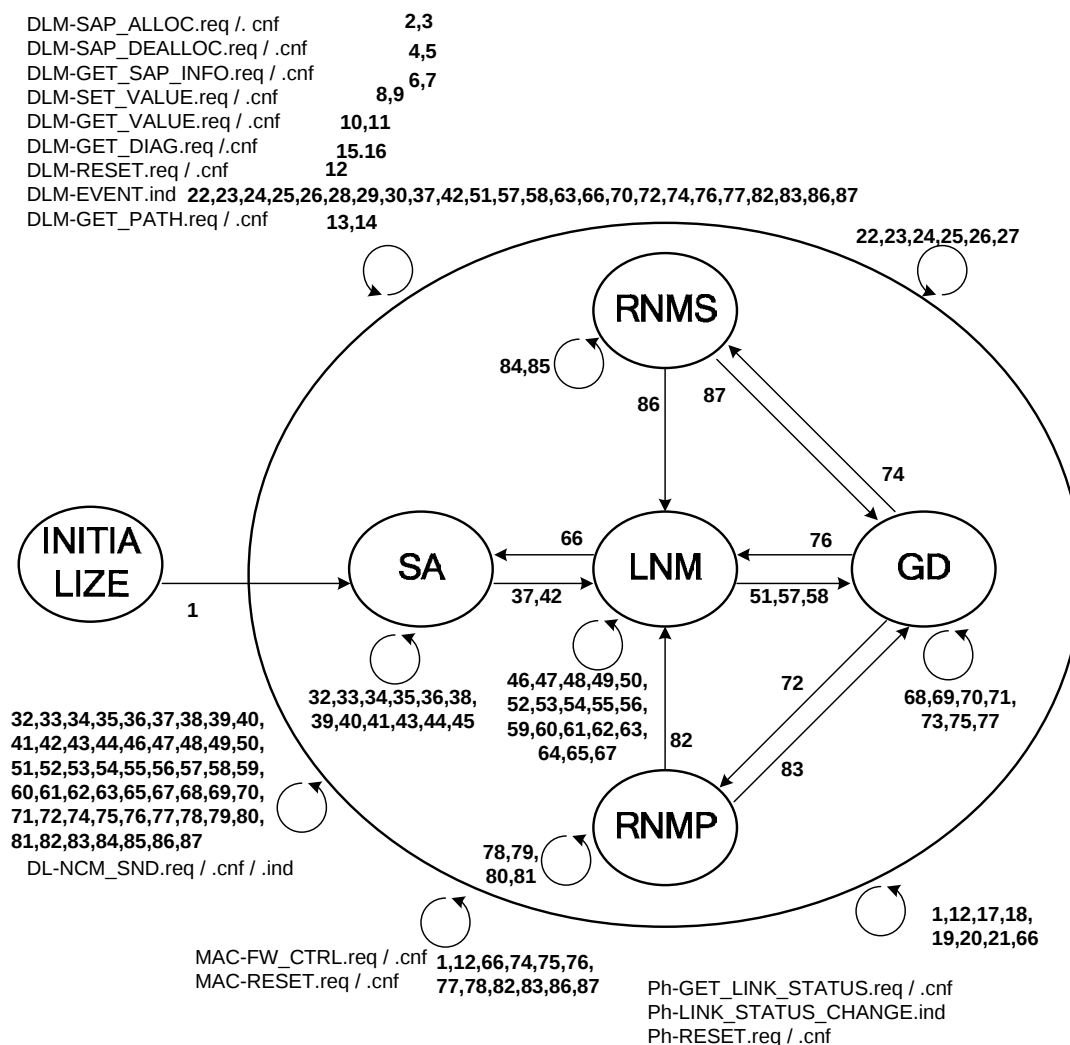


Figure 22 – State transition diagram of DLM

SA (standalone)

This state means that the local initialization procedures have been successfully completed and the device is ready to be linked to the other devices. In the SA state, the DLM tries to find the other devices on the network. When a link is established, the state changes to the LNM state.

LNM (line network manager)

This state means that the local device is located at the end of a line network. The LNM device is linked on the line network with one of its two R-ports. Both frame forward functions are disabled in the LNM device.

GD (general device)

This state means that the local device is connected to the network through both its R-ports. Both frame forward functions are enabled in the GD device. However, the frame forward functions in the GD device are suspended until the device receives NCM_LS DLPDU or NCM_RS DLPDU.

RNMP (primary ring network manager)

This state means that the local device is automatically selected as the primary ring network manager in a ring network. The RNMP device selects one of its neighboring devices as the secondary ring network manager (RNMS) using the NCM_RS DLPDU. The RNMP device disables the frame forward function in the RNMS direction but keeps the frame forward function in the other direction enabled.

RNMS (secondary ring network manager)

This state means that the local device is selected as the secondary ring network manager in a ring network. The RNMS device disables the frame forward function in the RNMP direction but keeps the frame forward function in the other direction enabled.

Table 70 – DLM state table

#	Current	Event / Condition =>actions	Next state
1	INITIALIZE	POWER-ON or RESET / => INIT_ENV_VAR() INIT_SAP_INFO() INIT_DEV_INFO() INIT_NET_INFO() INIT_PATH_INFO() MAC-RESET.req { } Ph-RESET.req { } SET_BLOCK_PORT(INVALID_R_PORT) CLEAR_PORT_INFO(R-port1) CLEAR_PORT_INFO(R-port2)	SA
		Events triggered by the DL-Management service are listed below.	
2	Any state	DLM-SAP_ALLOC.req {SAP, DLS-user ID} / CHECK_ALLOC_SAP(SAP) = "True" => ALLOC_SAP(SAP, DLS-user ID) Status := "success" DLM-SAP_ALLOC.cnf {Status}	Any state
3	Any state	DLM-SAP_ALLOC.req {SAP, DLS-user ID} / CHECK_ALLOC_SAP(SAP) <> "True" => Status := "Failure – SAP is already allocated to another DLS-user" DLM-SAP_ALLOC.cnf {Status}	Any state
4	Any state	DLM-SAP_DEALLOC.req {SAP} / CHECK_DEALLOC_SAP(SAP) = "True" => DEALLOC_SAP(SAP) Status := "success" DLM-SAP_DEALLOC.cnf {Status}	Any state
5	Any state	DLM-SAP_DEALLOC.req {SAP} / CHECK_DEALLOC_SAP(SAP) <> "True" => Status := "Failure – SAP is not allocated to a DLS-user" DLM-SAP_DEALLOC.cnf {Status}	Any state
6	Any state	DLM-GET_SAP_INFO.req {SAP} / CHECK_ALLOCEDSAP(SAP) = "True" => DLS-user ID := GET_USERID_FOR_SAP(SAP) Status := "Success" DLM-GET_SAP_INFO.cnf {Status, DLS-user ID}	Any state

#	Current	Event / Condition =>actions	Next state
7	Any state	DLM-GET_SAP_INFO.req {SAP} / CHECK_ALLOCEDSAP(SAP) <> "True" => DLS-user ID := INVALID_USER_ID Status := "Failure – SAP is not allocated to a DLS-user" DLM-GET_SAP_INFO.cnf {Status, DLS-user ID}	Any state
8	Any state	DLM-SET_VALUE.req {variable name, desired value} / CHECK_VALUE(variable name, desired value) = "valid" => SET_VALUE(variable name, desired value) Status := "success" DLM-SET_VALUE.cnf {Status }	Any state
9	Any state	DLM-SET_VALUE.req {variable name, desired value} / CHECK_VALUE(variable name, desired value) <> "valid" => Status := "Failure – invalid parameters in the request" DLM-SET_VALUE.cnf {Status }	Any state
10	Any state	DLM-GET_VALUE.req {variable name} / CHECK_VAR(variable name) = "valid" => Current value := GET_CURRENT_VAL(variable name) Status := "success" DLM-GET_VALUE.cnf {Status , Current value }	Any state
11	Any state	DLM-GET_VALUE.req {variable name} / CHECK_VAR(variable name) <> "valid" => Current value := INVALID_VALUE Status := "Failure – invalid parameters in the request" DLM-GET_VALUE.cnf {Status , Current value }	Any state

#	Current	Event / Condition =>actions	Next state
12	Any state	DLM-RESET.req { } / => INIT_ENV_VAR() INIT_SAP_INFO() INIT_DEV_INFO() INIT_NET_INFO() INIT_PATH_INFO() MAC-RESET.req { } Ph-RESET.req { } SET_BLOCK_PORT(INVALID_R_PORT) CLEAR_PORT_INFO(R-port1) CLEAR_PORT_INFO(R-port2) Status := "success" DLM-RESET.cnf {Status}	SA
13	Any state	DLM-GET_PATH.req {Addr} / CHECK_ADDR(Addr) = "valid" => R-port := GET_DST_PORT(addr) D_MAC_addr := GET_DST_MAC_ADDR(addr) Status := "success" DLM-GET_PATH.cnf {Status, R-port, D_MAC_Addr}	Any state
14	Any state	DLM-GET_PATH.req {Addr} / CHECK_ADDR(Addr) <> "valid" => R-port := INVALID_R_PORT D_MAC_addr := INVALID_MAC_ADDR Status := "Failure – invalid parameters in the request" DLM-GET_PATH.cnf {Status, R-port, D_MAC_Addr}	Any state
15	Any state	DLM-GET_DIAG.req {Diag-type, addr} / CHECK_DIAG_TYPE(Diag-type, addr) = "True" => Status := "success" Diag-info := GET_DIAG_INFO(Diag-type) DLM-GET_DIAG.cnf {Status, Diag-info}	Any state
16	Any state	DLM-GET_DIAG.req {Diag-type, addr} / CHECK_DIAG_TYPE (Diag-type, addr) <> "True" => Status := "Failure – invalid parameters in the request" DLM-GET_DIAG.cnf {Status, Diag-info}	Any state
		Events triggered by the DLM event generator are listed below.	

#	Current	Event / Condition =>actions	Next state
17	Any state	Ph_LINK_STATUS_CHANGE.ind {R-port} / => Ph-GET_LINK_STATUS.req {R-port}	Any state
18	Any state	Ph-GET_LINK_STATUS.cnf {R-port, L_status} / L_status = "link active" && CHECK_NEWLY_LINK_ACTV(R-port) = "True" => NEWLY_LINK_ACTV(R-port)	Any state
19	Any state	Ph-GET_LINK_STATUS.cnf {R-port, L_status} / L_status = "link active" && CHECK_NEWLY_LINK_ACTV(R-port) <> "True" => (none)	Any state
20	Any state	Ph-GET_LINK_STATUS.cnf {R-port, L_status} / L_status = "link inactive" && CHECK_NEWLY_LINK_INACTV(R-port) = "True" => NEWLY_LINK_INACTV(R-port)	Any state
21	Any state	Ph-GET_LINK_STATUS.cnf {R-port, L_status} / L_status = "link inactive" && CHECK_NEWLY_LINK_INACTV(R-port) <> "True" => (none)	Any state
		Device information-related procedures are listed below.	
22	Any state	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) <> "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) = "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) = "True" => events := EVENT_THIS_ADDR_COLLISION EVENT_IN_DEVICE DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Any state
23	Any state	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) = "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) <> "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) = "True" => events := EVENT_NET_ADDR_COLLISION EVENT_IN_DEVICE DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Any state

#	Current	Event / Condition =>actions	Next state
24	Any state	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) = "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) <> "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) <> "True" => events := EVENT_NET_ADDR_COLLISION DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Any state
25	Any state	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) <> "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) = "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) <> "True" => events := EVENT_THIS_ADDR_COLLISION DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Any state
26	Any state	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) <> "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) <> "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) = "True" => events := EVENT_IN_DEVICE DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Any state
27	Any state	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) <> "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) <> "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) <> "True" => SAVE_DEV_INFO(DLMDU, length, R-port)	Any state
		Path table update procedures are listed below	
28	Any state	DELETE_DEV_INFO(Device UID) / CHECK_UID(UID) = "valid" && CHECK_THIS_ADDR_COLLISION_CLEAR(UID) <> "True" && CHECK_NET_ADDR_COLLISION_CLEAR(UID) <> "True" => DEL_DEV(UID) Events := EVENT_OUT_DEVICE DLM-EVENT.ind(Events)	Any state

#	Current	Event / Condition =>actions	Next state
29	Any state	DELETE_DEV_INFO(Device UID) / CHECK_UID(UID) = "valid" && CHECK_THIS_ADDR_COLLISION_CLEAR(UID) = "True" && CHECK_NET_ADDR_COLLISION_CLEAR(UID) <> "True" => DEL_DEV(UID) Events := EVENT_OUT_DEVICE EVENT_THIS_ADDR_COLLISION_CLEAR DLM-EVENT.ind(Events)	Any state
30	Any state	DELETE_DEV_INFO(Device UID) / CHECK_UID(UID) = "valid" && CHECK_THIS_ADDR_COLLISION_CLEAR(UID) <> "True" && CHECK_NET_ADDR_COLLISION_CLEAR(UID) = "True" => DEL_DEV(UID) Events := EVENT_OUT_DEVICE EVENT_NET_ADDR_COLLISION_CLEAR DLM-EVENT.ind(Events)	Any state
31	Any state	DELETE_DEV_INFO(Device UID) / CHECK_UID(UID) <> "valid" => (<none>)	Any state
		State transitions are listed below.	
32	SA	NEWLY_LINK_ACTV(R-port) / => Block-R-port := INVERT_PORT(R-port) SET_BLOCK_PORT(Block-R-port) START_TIMER(FamilyReqT) DST_addr := C_NCM_ADDR NCMT := NCM_FAMILY_REQ DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
33	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_FAMILY_REQ => SAVE_UID_RP(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length) DST_addr := C_NCM_ADDR NCMT := NCM_FAMILY_RES DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA

#	Current	Event / Condition =>actions	Next state
34	SA	Timer(FamilyReqT) expired / => START_TIMER(FamilyReqT) DST_addr := C_NCM_ADDR NCMT := NCM_FAMILY_REQ DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
35	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_FAMILY_RES => STOP_TIMER(FamilyReqT) SAVE_UID_RP(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM_FAMILY) START_TIMER(MediaLinkedT) DST_addr := C_NCM_ADDR NCMT := NCM_MEDIA_LINKED DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
36	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "False" => (<none>)	SA

#	Current	Event / Condition =>actions	Next state
37	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ML) = "True" => SET_BLOCK_PORT(INVALID_R_PORT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) Events := EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr := C_NCM_ADDR NCMT := NCM_ADV_THIS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} DST_addr := C_NCM_ADDR NCMT := NCM_LINE_START DLMDU := local device information & network information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNM
38	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ML) = "False" => STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ADV) DST_addr := C_NCM_ADDR NCMT := NCM_ADV_THIS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA

#	Current	Event / Condition =>actions	Next state
39	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "False" => STORE_DEV_INFO(DLMDU, Length) DST_addr := C_NCM_ADDR NCMT := NCM_ADV_THIS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
40	SA	Timer(MediaLinkedT) expired / => START_TIMER(MediaLinkedT) DST_addr := C_NCM_ADDR NCMT := NCM_MEDIA_LINKED DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
41	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "False" => (<none>)	SA

#	Current	Event / Condition =>actions	Next state
42	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "True" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "True" CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "True") => STOP_TIMER(MediaLinkedT) SET_BLOCK_PORT(INVALID_R_PORT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) Events := EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr := C_NCM_ADDR NCMT := NCM_LINE_START DLMDU := local device information & network information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNM
43	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "False" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "True" CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "True") => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ML)	SA
44	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "False" && CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "False") => STORE_DEV_INFO(DLMDU, Length)	SA

#	Current	Event / Condition =>actions	Next state
45	SA	NEWLY_LINK_INACTV(R-port) / => INIT_DEV_INFO() INIT_NET_INFO() INIT_PATH_INFO() SET_BLOCK_PORT(INVALID_R_PORT) CLEAR_PORT_INFO(R-port1) CLEAR_PORT_INFO(R-port2)	SA
46	LNМ	NEWLY_LINK_ACTV(R-port) / => START_TIMER(FamilyReqT) DST_addr := C_NCM_ADDR NCMT := NCM_FAMILY_REQ DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
47	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_FAMILY_REQ => SAVE_UID_RP(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length) DST_addr := C_NCM_ADDR NCMT := NCM_FAMILY_RES DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
48	LNМ	Timer(FamilyReqT) expired / => START_TIMER(FamilyReqT) DST_addr := C_NCM_ADDR NCMT := NCM_FAMILY_REQ DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ

#	Current	Event / Condition =>actions	Next state
49	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_FAMILY_RES => STOP_TIMER(FamilyReqT) SAVE_UID_RP(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM_FAMILY) START_TIMER(MediaLinkedT) DST_addr := C_NCM_ADDR NCMT := NCM_MEDIA_LINKED DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
50	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "False" => (<none>)	LNМ
51	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ML) = "True" => STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHAGNE_DLM_STATE(GD) Events := EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) CLEAR_LNM_UID(R-port) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr := C_NCM_ADDR NCMT := NCM_ADV_THIS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	GD

#	Current	Event / Condition =>actions	Next state
52	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ML) = "False" => STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ADV) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr := C_NCM_ADDR NCMT := NCM_ADV_THIS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
53	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "False" && CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr := C_NCM_ADDR NCMT := NCM_ADV_THIS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
54	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_ECHO(DLMDU, Length) = "True" => (<none>)	LNМ

#	Current	Event / Condition =>actions	Next state
55	LNМ	Timer(MediaLinkedT) expired / => START_TIMER(MediaLinkedT) DST_addr := C_NCM_ADDR NCMT := NCM_MEDIA_LINKED DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
56	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "False" => (<none>)	LNМ
57	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "True" && CHECK_LNM_UID(DLMDU) = "True" && CHECK_UID_RP(DLMDU, R-port) = "True" => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHAGNE_DLM_STATE(GD) Events := EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) CLEAR_LNM_UID(R-port) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	GD

#	Current	Event / Condition =>actions	Next state
58	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "True" && CHECK_LNM_UID(DLMDU) = "False" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "True" CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "True") => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHAGNE_DLM_STATE(GD) Events := EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) CLEAR_LNM_UID(R-port) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	GD
59	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "False" && CHECK_LNM_UID(DLMDU) = "True" && CHECK_UID_RP(DLMDU, R-port) = "True" => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ML) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	LNМ

#	Current	Event / Condition =>actions	Next state
60	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "False" && CHECK_LNM_UID(DLMDU) = "False" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "True" CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "True") => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ML) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	LNМ
61	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_LNM_UID(DLMDU) = "True" && CHECK_UID_RP(DLMDU, R-port) = "False" && CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	LNМ
62	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_LNM_UID(DLMDU) = "False" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "False" && CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "False") CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	LNМ

#	Current	Event / Condition =>actions	Next state
63	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_ECHO(DLMDU, Length) = "True" => CHANGE_TOPOLOGY(NET_TPG_RING) Events := EVENT_NET_TPG_CHG DLM-EVENT.ind(events) START_TIMER(ChangeRingStateT)	LNМ
64	LNМ	Timer(ChangeRingStateT) expired / => START_TIMER(ChangeRingStateT)	LNМ
65	LNМ	NEWLY_LINK_INACTV(R-port) / CHECK_PORT_INFO(R-port, PORT_CFM) = "False" => DELETE_PATH_INFO(R-port) CLEAR_PORT_INFO(R-port) DST_addr := C_NCM_ADDR NCMT := NCM_LINE_START DLMDU := local device information & network information Length := size of DLMDU R-port := INVERT_PORT(R-port) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
66	LNМ	NEWLY_LINK_INACTV(R-port) / CHECK_PORT_INFO(R-port, PORT_CFM) = "True" => CHANGE_TOPOLOGY(NET_TPG_SA) CHAGNE_DLM_STATE(SA) Events := EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) UPDATE_PATH_TABLE(SA) INIT_DEV_INFO() INIT_NET_INFO() SET_BLOCK_PORT(INVALID_R_PORT) CLEAR_PORT_INFO(R-port1) CLEAR_PORT_INFO(R-port2) MAC-RESET.req { } Ph-RESET.req { }	SA

#	Current	Event / Condition =>actions	Next state
67	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_LINE_START => CHANGE_LNM_UID(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length)	LNМ
68	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_ECHO(DLMDU, Length) = "True" => (<none>)	GD
69	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr := C_NCM_ADDR NCMT := NCM_ADV_THIS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	GD
70	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_ECHO(DLMDU, Length) = "True" => CHANGE_TOPOLOGY(NET_TPG_RING) Events := EVENT_NET_TPG_CHG DLM-EVENT.ind(events) START_TIMER(ChangeRingStateT)	GD
71	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	GD

#	Current	Event / Condition =>actions	Next state
72	GD	Timer(ChangeRingStateT) expired / CHECK_NET_TOPOLOGY() = NET_TPG_RING && CHECK_RNMP() = RNMP => STOP_TIMER(ChangeRingStateT) CHAGNE_DLM_STATE(RNMP) Events := EVENT_DEV_STATE_CHG DLM-EVENT.ind (events) CLEAR_LNM_UID(R-port1) CLEAR_LNM_UID(R-port2) DST_addr := C_NCM_ADDR NCMT := NCM_RING_START DLMDU := local device information & network information Length := size of DLMDU Rnms_UID := GET_RNMS_UID() SET_RNMPS_UID(Rnms_UID, DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port2} START_TIMER(AckRNMT)	RNMP
73	GD	Timer(ChangeRingStateT) expired / CHECK_NET_TOPOLOGY() = NET_TPG_RING && CHECK_RNMP() <> RNMP => STOP_TIMER(ChangeRingStateT)	GD

#	Current	Event / Condition =>actions	Next state
74	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RING_START && GET_NET_RNMS_UID(DLMDU) = device information.UID => STORE_DEV_INFO(DLMDU, Length) CHAGNE_DLM_STATE(RNMS) Events := EVENT_DEV_STATE_CHG DLM-EVENT.ind (events) CLEAR_LNM_UID(R-port1) CLEAR_LNM_UID(R-port2) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr := SRC_addr NCMT := NCM_ACK_RNMS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} Forward-control := FW_DISABLE R-port := GET_R_PORT_FOR(addr of RNMP) FW_CTRL(R-port, Forward-control) R-port := INVERT_PORT(GET_R_PORT_FOR(addr of RNMP)) Forward-control := FW_ENABLE FW_CTRL(R-port, Forward-control)	RNMS
75	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RING_START && GET_NET_RNMS_UID(DLMDU) <> device information.UID => STORE_DEV_INFO(DLMDU, Length) CLEAR_LNM_UID(R-port1) CLEAR_LNM_UID(R-port2) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} Forward_control := FW_ENABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	GD

#	Current	Event / Condition =>actions	Next state
76	GD	NEWLY_LINK_INACTV(R-port) / => DELETE_PATH_INFO(R-port) CLEAR_PORT_INFO(R-port) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) Events := EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr := C_NCM_ADDR NCMT := NCM_LINE_START DLMDU := local device information & network information Length := size of DLMDU R-port := INVERT_PORT(R-port) DL-NCM_SND.req(DST_addr, NCMT, DLMDU, Length, R-port) UPDATE_PATH_TABLE(LINE) Forward_control := FW_DISABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	LNM
77	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_LINE_START => STORE_DEV_INFO(DLMDU, Length) CHANGE_TOPOLOGY(NET_TPG_LINE) Events := EVENT_NET_TPG_CHG DLM-EVENT.ind (events) UPDATE_PATH_TABLE(LINE) CHANGE_LNM_UID(DLMDU, R-port) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} Forward_control := FW_ENABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	GD

#	Current	Event / Condition =>actions	Next state
78	RNMP	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ACK_RNMS && CHECK_RNMS_UID(DLMDU) = "True" => STOP_TIMER(AckRNMST) Forward-control := FW_DISABLE R-port := GET_R_PORT_FOR(addr of RNMS) FW_CTRL(R-port, Forward-control) R-port := INVERT_PORT(GET_R_PORT_FOR(addr of RNMS)) Forward-control := FW_ENABLE FW_CTRL(R-port, Forward-control)	RNMP
79	RNMP	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ACK_RNMS && CHECK_RNMS_UID(DLMDU) = "False" => (<none>)	RNMP
80	RNMP	Timer(AckRNMST) expired / => DST_addr := addr of RNMS NCMT := NCM_RETRY_RNMS DLMDU := local device information Length := size of DLMDU R-port := both R-port DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} START_TIMER(AckRNMST)	RNMP
81	RNMP	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RING_START && CHECK_ECHO(DLMDU, Length) = "True" => (<none>)	RNMP

#	Current	Event / Condition =>actions	Next state
82	RNMP	NEWLY_LINK_INACTV(R-port) / => DELETE_PATH_INFO(R-port) CLEAR_PORT_INFO(R-port) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) Events := EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr := C_NCM_ADDR NCMT := NCM_LINE_START DLMDU := local device information & network information Length := size of DLMDU R-port := INVERT_PORT(R-port) DL-NCM_SND.req(DST_addr, NCMT, DLMDU, Length, R-port) UPDATE_PATH_TABLE(LINE) Forward_control := FW_DISABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	LNM
83	RNMP	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_LINE_START => STORE_DEV_INFO(DLMDU, Length) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(GD) Events := EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) UPDATE_PATH_TABLE(LINE) CHANGE_LNM_UID(DLMDU, R-port) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} Forward_control := FW_ENABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	GD

#	Current	Event / Condition =>actions	Next state
84	RNMS	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RING_START && GET_NET_RNMS_UID(DLMDU) = device information.UID => STORE_DEV_INFO(DLMDU, Length) CLEAR_LNM_UID(R-port1) CLEAR_LNM_UID(R-port2) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr := SRC_addr NCMT := NCM_ACK_RNMS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	RNMS
85	RNMS	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RETRY_RNMS => STORE_DEV_INFO(DLMDU, Length) DST_addr := SRC_addr NCMT := NCM_ACK_RNMS DLMDU := local device information Length := size of DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	RNMS
86	RNMS	NEWLY_LINK_INACTV(R-port) / => DELETE_PATH_INFO(R-port) CLEAR_PORT_INFO(R-port) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) Events := EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr := C_NCM_ADDR NCMT := NCM_LINE_START DLMDU := local device information & network information Length := size of DLMDU R-port := INVERT_PORT(R-port) DL-NCM_SND.req(DST_addr, NCMT, DLMDU, Length, R-port) UPDATE_PATH_TABLE(LINE) Forward_control := FW_DISABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	LNM

#	Current	Event / Condition =>actions	Next state
87	RNMS	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_LINE_START => STORE_DEV_INFO(DLMDU, Length) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(GD) Events := EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) UPDATE_PATH_TABLE(LINE) CHANGE_LNM_UID(DLMDU, R-port) Forward-R-port := INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} Forward_control := FW_ENABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	GD

7.3.4 DLM functions

Table 71 shows the internal functions provided by the DLM.

Table 71 – DLM function table

Function name	Input	Output	Operation
INIT_ENV_VAR			Initialize DLL configuration parameters (see 4.6.2).
INIT_SAP_INFO			Initialize SAP information (see 4.6.4).
INIT_DEV_INFO			Initialize local device information (see 4.6.5).
INIT_NET_INFO			Initialize network information (see 4.6.6).
INIT_PATH_INFO			Initialize path table information (see 4.6.7).
CHECK_ALLOC_SAP	SAP	True/False	Return TRUE if the SAP is available. Otherwise, return FALSE.
ALLOC_SAP	SAP DLS-user ID		Register DLS-user ID in the SAP information.
CHECK_DEALLOC_SAP	SAP	True/ False	Check if the SAP can be deallocated.
DEALLOC_SAP	SAP		Remove the DLS-user ID from the SAP information.
CHECK_ALLOCEDSAP	SAP	True/ False	Check if the SAP is already allocated to a certain DLS-user.
GET_USERID_FOR_SAP	SAP	DLS-user ID	Return the DLS-user ID that owns the allocated SAP.
CHECK_VALUE	Variable name desired value	Valid/Invalid	Check that the requested variables with desired value are valid.
SET_VALUE	Variable name desired value		Set the value of the requested variable.
GET_CURRENT_VAL	Variable name	Current_value	Get the value of the requested variable.
CHECK_ADDR	DL– entity identifier	Valid/Invalid	Check if the designated DL– entity identifier is valid. Return “Valid” if the unicast DL– entity identifier is registered in the path table. Return “Invalid” if the unicast DL– entity identifier is not registered in the path table. Return “Valid” if the designated DL– entity identifier is not in the range of the unicast DL– entity identifier.
GET_DST_PORT	DL– entity identifier	R-port	Return the destination R-port for the designated device in the path table.
GET_DST_MAC_ADDR	Address	MAC address	Return the ISO/IEC 8802-3:2000 MAC address for the designated device in the path table.
CHECK_DIAG_TYPE	Diagnostic_type Addr	True/False	Check if the diagnostic information type from the DLS-user is available.
GET_DIAG_INFO	Diagnostic_type Addr	Diagnostic information	Return the diagnostic information according to the diagnostic information type.
CHECK_NEWLY_LINK_A CTV	R-port	True/False	Check if the R-port has changed from “link Inactive” to “link Active” status.
CHECK_NEWLY_LINK_I NACTV	R-port	True/False	Check if the R-port has changed from “link Active” to link Inactive” status.

Function name	Input	Output	Operation
UPDATE_PORT_INFO	R-port status		Update the port information in the device information (see 4.6.5.9).
CLEAR_PORT_INFO	R-port		Clear the port information of R-port.
CHECK_PORT_INFO	R-port status	True/False	Check if the port information bit of R-port in the device information is set. Return TRUE if the status bit in the port information of R-port is set. Otherwise, return FALSE.
SET_BLOCK_PORT	R-port		Set R-port to blocked state. Set R-port1 and R-port2 to active state if R-port is INVALID_R_PORT. Otherwise, set R-port to blocked state.
CHECK_ECHO	DLMDU length	True/False	Check if the DLPDU was generated by the device itself.
INC_HOP_CNT	DLMDU		Increment the Hop count field in the DLMDU.
INVERT_PORT	R-port	R-port	Invert logical interface to the R-port. Return R-port2 if the input is R-port1 and return R-port1 if the input is R-port2.
CHECK_NET_TOPOLOGY		Network topology	Check if the network is configured as a ring topology. Return NET_TPG_RING if the broadcast network control message is received by the device itself. Otherwise, return NET_TPG_LINE.
CHECK_NET_ADDR_COLLISION	DLMDU	True/False	Return TRUE if the source DL- entity identifier in the DLMDU is duplicated at one of the other devices on the network. Otherwise, return FALSE.
CHECK_THIS_ADDR_COLLISION	DLMDU		Return TRUE if the source DL- entity identifier in the DLMDU is duplicated in the local device itself. Otherwise, return FALSE.
CHECK_NEWLY_IN_DEVICE	DLMDU	True/False	Check if the source DL- entity identifier in the DLMDU is not registered in the Device information.
SAVE_DEV_INFO	DLMDU length R-port		Store the device information in the DLMDU to the path table.
DEL_DEV	Device UID		Remove the path table item corresponding to the Device UID.
DELETE_PATH_INFO	R-port		Remove the path table item corresponding to the R-port.
CHANGE_TOPOLOGY	Topology		Update the topology in the network information.
CHANGE_LNM_UID	DLMDU R-port		Store the source DL- entity identifier in the DLMDU to the LNM device UID for R-port in the network information.
CLEAR_LNM_UID	R-port		Clear the LNM device UID for R-port in the network information.
CHAGNE_DLM_STATE	DLM-state		Update the DLS state in the device information.
SAVE_UID_RP	DLMDU R-port		Store the source DL- entity identifier in the DLMDU to the Device UID for R-port in the device information.

Function name	Input	Output	Operation
CHECK_UID_RP	DLMDU R-port	True/False	Return TRUE if the source DL– entity identifier in the DLMDU has been registered as Device UID for R-port in the device information. Otherwise, return FALSE.
CHECK_NET_PORT_INFO	DLMDU status	True/False	Check if the port information bit in the DLMDU is set. Return TRUE if the status bit in the port information of the DLMDU is set. Otherwise, return FALSE.
CHECK_LNM_UID	DLMDU	True/False	Return TRUE if the source DL– entity identifier in the DLMDU has been registered as LNM in the network information. Otherwise, return FALSE.
UPDATE_PATH_TABLE	Topology		Update and recalculate the destination port in the path table according to the network topology.
CHECK_LNMS		True/False	Check if two LNMs are listed in the path table.
FW_CTRL	R-port Control		Enable or disable the frame forward function for the R-port.
GET_RNMS_UID		UID	Return the device UID registered as RNMS in the path table. Return INVALID_UID if the RNMS is not found in the path table.
GET_NET_RNMS_UID	DLMDU	UID	Return the RNMS_device UID in the DLMDU.
CHECK_RNMS_UID	DLMDU	True/False	Return TRUE if the source DL– entity identifier in the DLMDU is the same as the RNMS device UID in the network information. Otherwise, return FALSE.
GET_R_PORT_FOR	DL– entity identifier	R-port	Return the destination R-port to the DL– entity identifier from the path table. Return INVALID_ADDR if no table entry is found.
START_TIMER	desired time		Start the timer.
STOP_TIMER	desired time		Stop the timer.

8 Constants and error codes

8.1 General

This clause describes the Type 21 DLL constants and their values.

8.2 Constants

Table 72 summarizes all DLL constants.

Table 72 – DLL constants

Constant	Value	Description
C_HIGEST_PRIORITY	3	The highest message priority
INVALID_USER_ID	0	Invalid user identification
INVALID_R_PORT	0	Invalid R-port
NCM_FAMILY_REQ	1	“link active req.” network control message
NCM_FAMILY_RES	2	“link active res.” network control message
NCM_MEDIA_LINKED	3	“indicate established Type 21 link” network control message
NCM_ADV_THIS	4	“advertise this information” network control message
NCM_LINE_START	5	“starting line topology network” network control message
NCM_RING_START	6	“starting ring topology network” network control message
NCM_ACK_RNMS	7	ACK message from RNMS to RNMP
NCM_RETRY_RNMS	8	REQ message from RNMP to RNMS
EVENT_NET_TPG_CHG	0x01	Event generated when the network topology is changed
EVENT_DEV_STATE_CHG	0x02	Event generated when the DLM state is changed
EVENT_THIS_ADDR_COLLISION	0x04	Event generated when the local DL– entity identifier collision is detected
EVENT_THIS_ADDR_COLLISION_CLEAR	0x08	Event generated when the local DL– entity identifier collision is cleared
EVENT_NET_ADDR_COLLISION	0x10	Event generated when the network DL– entity identifier collision is detected
EVENT_NET_ADDR_COLLISION_CLEAR	0x20	Event generated when the network DL– entity identifier collision is cleared
EVENT_IN_DEVICE	0x40	Event generated when a device first joins the network
EVENT_OUT_DEVICE	0x80	Event generated when a device disconnects from the network
C_NCM_ADDR	0xFFFE	The DL– entity identifier for network control messages
C_BROADCAST_ADDR	0xFFFF	DL– entity identifier for broadcast
FW_ENABLE	1	Enable frame forwarding function
FW_DISABLE	0	Disable frame forwarding function
INVALID_UID	0	Invalid unique identification
INVALID_ADDR	0xFFDC	Invalid address
PORT_LINK_DOWN	0x01	R-port link down
PORT_CFM_FAMILY	0x02	Received NCM_FAMILY_RES frame from R-port
PORT_WAIT_ADV	0x04	Waiting for NCM_ADV_THIS frame from R-port
PORT_WAIT_ML	0x08	Waiting for NCM_MEDIA_LINKED frame from R-port
PORT_CFM	0x10	Device state confirm
PORT1_LINK_DOWN	0x0001	R-port1 link down
PORT2_LINK_DOWN	0x0100	R-port2 link down
FamilyReqT	100 ms	NCM_FAMILY_RES frame waiting time
MediaLinkedT	100 ms	NCM_ADV_THIS frame waiting time
AckRN MST	100 ms	NCM_ACK_RNMS frame waiting time
ChangeRingStateT	1 ms	Ring state change timeout

8.3 Data-link layer error codes

Table 73 summarizes all DLL error codes.

Table 73 – Type 21 DLL error codes

Error	value	Description
ERR_FRAME_SIZE	0x1001	Frame size error
ERR_PROTOCOL_VER	0x1002	Protocol version error
ERR_NOT4ME_FRAME	0x1003	The received frame is not intended for the local device
ERR_SAP	0x1004	Service access point error
ERR_SERVICE_CMD	0x1005	Service command error
ERR_NO_DST_ADDR	0x1006	No device found matching the destination DL– entity identifier
ERR_PRM	0x1007	Invalid parameter
ERR_NG_DEV_INFO	0x1008	Device information error
ERR_GET_DEV_INFO	0x1009	Device information size error
ERR_Q_ALREADY_FULL	0x100A	Queue is full
ERR_SAP_ALREADY_REGED	0x100B	The SAP is already allocated to a DLS-user.
ERR_SAP_NOT_REGED	0x100C	The SAP is not allocated to a DLS-user
ERR_NO_MATCHED_ITEM	0x100D	No item is found in the NMIB.
ERR_INVALID_DIAG_TYPE	0x100E	Diagnostic information type is invalid
ERR_DEV_CNT_EXIED	0x2001	DL– entity identifier is bigger than MAX_ADDR.
NOT_SUPPORTEDSAP	0x2002	Invalid SAP address error
INVALID_DEV_INFO	0x2003	Invalid device information error

Bibliography

IEC/TR 61158-1:2010², *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-2:2010², *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-5-21:2010², *Industrial communication networks – Fieldbus specifications – Part 5-21: Application layer service definition – Type 21 elements*

IEC 61158-6-21:2010², *Industrial communication networks – Fieldbus specifications – Part 6-21: Application layer protocol specification – Type 21 elements*

IEC 61588, *Precision clock synchronization protocol for networked measurement and control systems*

ISO/IEC TR8802–1, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 1: Overview of Local Area Network Standards*

IETF RFC 768, *User Datagram Protocol (UDP)*; available at <<http://www.ietf.org>>

IETF RFC 791, *Internet Protocol*; available at <<http://www.ietf.org>>

IETF RFC 793, *Transmission Control Protocol*, available at <<http://www.ietf.org>>

² To be published.

SOMMAIRE

AVANT-PROPOS.....	110
INTRODUCTION.....	112
1 Domaine d'application	114
1.1 Généralités.....	114
1.2 Spécifications.....	114
1.3 Procédures.....	114
1.4 Applicabilité.....	114
1.5 Conformité	115
2 Références normatives.....	115
3 Termes, définitions, symboles et abréviations.....	115
3.1 Termes et définitions du modèle de référence	115
3.2 Termes, définitions et conventions des services	117
3.3 Termes et définitions communs	118
3.4 Symboles et abréviations	122
4 Vue d'ensemble du protocole de liaison de données.....	123
4.1 Généralités.....	123
4.2 Vue d'ensemble du contrôle d'accès au support physique	123
4.3 Service pris en charge à partir de la couche physique	124
4.4 Architecture DLL	124
4.5 Type de données.....	127
4.6 Paramètres et variables locaux	130
5 Structure générale et codage.....	147
5.1 Vue d'ensemble.....	147
5.2 Structure et codage de MAPDU.....	147
5.3 Structure, codage et éléments de trame MAC commune.....	147
5.4 Ordre d'émission des bits	158
5.5 DLPDU non valide	158
6 Structure et procédure de DLPDU	158
6.1 Généralités.....	158
6.2 Champ commun aux DLPDU	159
6.3 Transfert de données DL.....	159
6.4 Transfert de DL-SPDATA (Données sporadiques de DL)	163
6.5 Messages de Contrôle du Réseau	165
7 Eléments de procédure de DLE	171
7.1 Structure globale	171
7.2 Machine protocolaire DL (DLPM).....	172
7.3 Protocole de gestion DLL	181
8 Constantes et codes d'erreur.....	214
8.1 Généralités.....	214
8.2 Constantes.....	214
8.3 Codes d'erreur de la couche Liaison de données.....	216
Bibliographie.....	217
Figure 1 – Relations entre DLSAP, adresses DLSAP et adresses DL de groupe	119
Figure 2 – Interaction des primitives de PhS avec une DLE	124

Figure 3 – Architecture de la couche de liaison de données.....	127
Figure 4 – Format de trame MAC commune pour la DLPDU de Type 21	148
Figure 5 – Format de trame MAC pour d'autres protocoles.....	148
Figure 6 – Champs version et longueur.....	150
Figure 7 – Champ DST_addr	151
Figure 8 – Champ SRC_addr	152
Figure 9 – Champ Contrôle de trame	152
Figure 10 – Champ Extension.....	155
Figure 11 – Champ DSAP	156
Figure 12 – Champ Point d'accès de service d'origine	156
Figure 13 – Longueur des informations de masque de groupe et d'extension	157
Figure 14 – Champ option de masque de groupe	158
Figure 15 – Champ commun aux DLPDU	159
Figure 16 – Construction d'une DLPDU de DT	160
Figure 17 – Structure d'une DLPDU de DT.....	160
Figure 18 – Structure de DLPDU de SPDT.....	163
Figure 19 – Structure de DLPDU NCM_LA.....	165
Figure 20 – Structure et éléments de DLL.....	172
Figure 21 – Diagramme des transitions d'états de la DLPM.....	175
Figure 22 – Diagramme des transitions d'états de la DLM.....	185
 Tableau 1 – Composants de la DLL	 125
Tableau 2 – Type de données UNSIGNEDn.....	128
Tableau 3 – Type de données INTEGERn.....	129
Tableau 4 – Paramètres de configuration DLE	130
Tableau 5 – Files d'attente pour la prise en charge du transfert de données	131
Tableau 6 – Variables de prise en charge de la gestion SAP	132
Tableau 7 – Variables de prise en charge de la gestion des informations du dispositif.....	133
Tableau 8 – Identifiant d'entité DL	133
Tableau 9 – Indicateurs de dispositif.....	134
Tableau 10 – Etat DLM	134
Tableau 11 – Identifiant unique de dispositif	134
Tableau 12 – Identifiant unique de dispositif connecté au Port R1.....	135
Tableau 13 – Identifiant unique de dispositif connecté au Port R2.....	135
Tableau 14 – Adresse MAC	135
Tableau 15 – Informations de port	136
Tableau 16 – Version de protocole.....	136
Tableau 17 – Type de dispositif	137
Tableau 18 – Description de dispositif	137
Tableau 19 – Nombre de sauts	137
Tableau 20 – Variables de prise en charge des informations de gestion du réseau	138
Tableau 21 – Topologie	138
Tableau 22 – Nombre de collisions	139

Tableau 23 – Nombre de dispositifs	139
Tableau 24 – Nombre de changements de topologie	139
Tableau 25 – Heure du dernier changement de topologie	139
Tableau 26 – UID de dispositif RNMP	140
Tableau 27 – UID de dispositif RNMS	140
Tableau 28 – UID du dispositif LNM pour le port R1.....	140
Tableau 29 – UID du dispositif LNM pour le port R2.....	141
Tableau 30 – Indicateurs de réseau	141
Tableau 31 – Variables et compteur de prise en charge de la gestion des informations de trajets	143
Tableau 32 – Nombre de sauts dans le sens du Port R1	144
Tableau 33 – Nombre de sauts dans le sens du Port R2	144
Tableau 34 – Port R préférentiel	144
Tableau 35 – Port R de destination	145
Tableau 36 – Nombre d'Entrées sur le réseau.....	146
Tableau 37 – Horodatage de l'Entrée sur le réseau.....	146
Tableau 38 – Nombre de sorties du réseau.....	146
Tableau 39 – Horodatage de la Sortie du réseau	146
Tableau 40 – Version et Longueur	150
Tableau 41 – Identifiant d'entité DL de destination.....	151
Tableau 42 – Identifiant d'entité DL d'origine	152
Tableau 43 – Contrôle de trame.....	153
Tableau 44 – Extension	155
Tableau 45 – Point d'accès de service de destination	156
Tableau 46 – Point d'accès de service d'origine	157
Tableau 47 – Paramètres de DLPDU de DT	161
Tableau 48 – Primitives échangées entre un utilisateur DLS et une DLE pour envoyer une DLPDU de DT	162
Tableau 49 – Primitives échangées entre un utilisateur DLS et des DLE pour recevoir une DLPDU de DT	163
Tableau 50 – Paramètres de DLPDU de SPDT	164
Tableau 51 – Primitives échangées entre un utilisateur DLS et des DLE pour envoyer une DLPDU de SPDT.....	164
Tableau 52 – Primitives échangées entre un utilisateur DLS et des DLE pour recevoir une DLPDU de SPDT.....	165
Tableau 53 – Paramètres de DLPDU NCM_LA.....	166
Tableau 54 – Paramètres de DLPDU NCM_AT	167
Tableau 55 – Paramètres de DLPDU NCM_LS.....	168
Tableau 56 – Paramètres de DLPDU NCM_RS	169
Tableau 57 – Paramètres de DLPDU NCM_AR	170
Tableau 58 – Primitives échangées entre DLPM et utilisateur DLS	173
Tableau 59 – Paramètres échangés entre DLPM et utilisateur DLS.....	173
Tableau 60 – Primitives échangées entre DLPM et DLM	174
Tableau 61 – Paramètres utilisés avec des primitives échangées entre DLPM et DLM.....	175
Tableau 62 – Table d'états DLPM	176

Tableau 63 – Table des fonctions DLPM	180
Tableau 64 – Primitives échangées entre DLM et utilisateur DLS	182
Tableau 65 – Paramètres utilisés avec des primitives échangées entre DLM et utilisateur DLS	183
Tableau 66 – Primitives échangées entre DLM et DMAC	183
Tableau 67 – Paramètres utilisés avec des primitives échangées entre DLM et DMAC	184
Tableau 68 – Primitives échangées entre DLM et DPHY	184
Tableau 69 – Paramètres utilisés avec des primitives échangées entre DLM et DPHY	184
Tableau 70 – Table d'états DLM	187
Tableau 71 – Table des fonctions DLM	212
Tableau 72 – Constantes de DLL	215
Tableau 73 – Codes d'erreur de DLL de Type 21	216

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 4-21: Spécification du protocole de la couche de liaison de données – Eléments de Type 21

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales ou régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.

La Norme internationale CEI 61158-4-21:2010 a été établie par le sous-comité 65C: Réseaux de communication industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

La présente norme annule et remplace la CEI/PAS 62573 publiée en 2008. Cette première édition constitue une révision technique.

La présente version bilingue publiée en 2012-01 correspond à la version anglaise monolingue publiée en 2010-08.

Le texte anglais de cette norme est issu des documents 65C/605/FDIS et 65C/619/RVD.

Le rapport de vote 65C/619/RVD donne toute information sur le vote ayant abouti à l'approbation de cette norme.

La version française de cette norme n'a pas été soumise au vote.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61158, présentées sous le titre général *Réseaux de communication industriels - Spécifications de bus de terrain*, est disponible sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

NOTE La révision de la présente norme sera synchronisée avec les autres parties de la série CEI 61158.

INTRODUCTION

Cette partie de la CEI 61158 fait partie d'une série élaborée pour faciliter l'interconnexion des composants de systèmes d'automatisation. Elle est apparentée à d'autres normes de cet ensemble, comme défini par le modèle de référence de bus de terrain "à trois couches" décrit dans la CEI/TR 61158-1.

Le protocole de liaison de données assure le service de liaison de données en utilisant les services disponibles à partir de la couche physique. Le principal objectif de la présente norme est de fournir un ensemble de règles de communication exprimées en termes de procédures à appliquer par des entités de liaison de données (DLE) homologues au cours de la communication. Ces règles de communication sont destinées à fournir une base saine de développement, de manière à satisfaire divers objectifs:

- a) servir de guide pour les ingénieurs d'application et les concepteurs;
- b) être utilisés pour les essais et l'acquisition d'équipements;
- c) servir de base, dans le cadre d'un accord donné, à l'admission de systèmes dans l'environnement OSI;
- d) approfondir les connaissances en matière de communications critiques du point de vue temporel (à priorité stricte) dans le cadre de l'OSI.

La présente norme couvre notamment la communication et l'interaction de capteurs, organes terminaux et autres dispositifs d'automatisation. L'utilisation de la présente norme, associée à d'autres normes qui font partie des modèles de référence OSI ou bus de terrain, permet de combiner et de faire fonctionner ensemble des systèmes qui seraient autrement incompatibles.

NOTE L'utilisation de certains des types de protocoles est restreinte par les détenteurs des droits de propriété intellectuelle correspondants. Quoi qu'il en soit, l'engagement pris par les détenteurs, quant à une diffusion limitée desdits droits de propriété intellectuelle, permet d'utiliser un type particulier de protocole de Couche Liaison de données avec des protocoles de Couche Physique et de Couche Application dans les combinaisons de types explicitement spécifiées dans les parties concernant les profils. L'utilisation des divers types de protocoles dans d'autres combinaisons peut nécessiter l'autorisation des détenteurs de leurs droits de propriété intellectuelle respectifs.

La Commission Electrotechnique Internationale (CEI) attire l'attention sur le fait qu'il est déclaré que la conformité au présent document peut impliquer l'utilisation de brevets relatifs aux éléments de Type 21 et éventuellement d'autres types indiqués dans 4.1, 4.2 et 7.3; ces brevets sont les suivants:

KR 0789444	[LS]	A communication packet processing apparatus and method for ring topology ethernet network capable of preventing permanent packet looping
KR 0732510	[LS]	Network system
KR 0870670	[LS]	Method for determining a Ring Manager Node

La CEI ne prend aucunement position en ce qui concerne la démonstration, la validité et l'étendue de ces droits de propriété.

Le détenteur de ces droits de propriété a donné l'assurance à la CEI qu'il consent à négocier des licences avec des demandeurs du monde entier, soit gratuitement, soit en des termes et à des conditions raisonnables et non discriminatoires. A ce propos, la déclaration du détenteur de ces droits de propriété est enregistrée à la CEI. Des informations peuvent être obtenues auprès de:

[LS]: LS Industrial Systems Co., Ltd.
 LS Tower 1026-6
 Hogle-dong, Dongan-gu,
 Anyang-si, Gyeonggi-do 431-848
 République de Corée

L'attention est attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues autres que ceux identifiés ci-dessus. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

L'ISO (www.iso.org/patents) et la CEI (http://www.iec.ch/tctools/patent_decl.htm) maintiennent des bases de données, consultables en ligne, des droits de propriété pertinents à leurs normes. Les utilisateurs sont encouragés à consulter ces bases de données pour obtenir les informations les plus récentes concernant les droits de propriété.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 4-21: Spécification du protocole de la couche de liaison de données – Eléments de Type 21

1 Domaine d'application

1.1 Généralités

La couche DLL permet la communication de données de base, critiques du point de vue temporel, entre dispositifs dans un environnement automatisé. Le Type 21 permet la communication de données cycliques et acycliques, fondée sur la priorité, en utilisant une technologie de commutation Ethernet interne à deux ports, en duplex intégral et exempte de collision. Pour diverses applications d'automatisation étendues, le Type 21 ne restreint pas la stratégie de planification cyclique/acyclique de la DLL.

1.2 Spécifications

La présente norme décrit:

- a) des procédures de transfert en temps opportun de données et d'informations de commande d'une entité utilisateur de liaison de données à une entité utilisateur homologue ainsi qu'entre entités de liaison de données qui constituent le fournisseur de services distribués de la liaison;
- b) des procédures offrant des moyens de communication fondées sur le MAC de l'ISO/CEI 8802-3, ainsi que des dispositions permettant d'ajouter et de retirer des nœuds au cours du fonctionnement normal;
- c) la structure des unités de données de protocole de liaison de données (DLPDU) de bus de terrain utilisées pour le transfert des données et les informations de commande par le protocole objet de la présente norme, ainsi que leur représentation en tant qu'unité de données d'interface de couche physique.

1.3 Procédures

Les procédures sont définies en termes:

- a) d'interactions entre Entités de Liaison de Données (DLE) homologues par échange de DLPDU de bus de terrain;
- b) d'interactions entre un prestataire de Service de Liaison de Données (DLS) et un utilisateur DLS au sein du même système, par échange de primitives DLS;
- c) d'interactions entre un fournisseur DLS et un fournisseur de Services de Couche Physique au sein du même système, par échange de primitives de services Ph;

1.4 Applicabilité

Ces procédures sont applicables à des instances de communication entre systèmes qui prennent en charge des services de communication à priorité stricte dans la couche de liaison de données de l'OSI ou des modèles de référence des bus de terrain et qui nécessitent la

faculté de s'interconnecter dans un environnement OSI (Interconnexion des systèmes ouverts). Les profils constituent un moyen simple, à attributs multiples, qui permet de résumer les capacités d'une mise en œuvre et par conséquent, son applicabilité à divers besoins de communication déterministes du point de vue temporel.

1.5 Conformité

La présente Norme spécifie également les conditions de conformité des systèmes mettant en œuvre ces procédures. La présente norme ne fournit pas d'essais destinés à démontrer la conformité à ces exigences.

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61158-3-21:2010¹, *Industrial Communication Networks – Fieldbus specifications – Part 3-21: Data-link layer service definition – Type 21 elements* (disponible uniquement en anglais)²

ISO/CEI 7498-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Le Modèle de base*

ISO/CEI 7498-3, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Dénomination et adressage*

ISO/IEC 8802-3:2000, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications* (disponible uniquement en anglais)

ISO/CEI 10731:1994, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base – Conventions pour la définition des services OSI*

3 Termes, définitions, symboles et abréviations

Pour les besoins du présent document, les termes, définitions, symboles, abréviations et conventions suivants s'appliquent.

3.1 Termes et définitions du modèle de référence

La présente norme repose en partie sur les concepts développés dans l'ISO/CEI 7498-1 et l'ISO/CEI 7498-3; elle utilise les termes suivants qui y sont définis.

3.1.1 adresse DL appelée	[ISO/CEI 7498-3]
3.1.2 adresse DL appelante	[ISO/CEI 7498-3]
3.1.3 connexion centralisée à points d'extrémité multiples	[ISO/CEI 7498-1]

¹ A publier.

² Les publications monolingues des séries IEC 61158 et IEC 61784 sont actuellement en cours de traduction.

3.1.4 (N)-entités correspondantes	[ISO/CEI 7498-1]
entités DL correspondantes (N=2)	
entités Ph correspondantes (N=1)	
3.1.5 démultiplexage	[ISO/CEI 7498-1]
3.1.6 adresse DL	[ISO/CEI 7498-3]
3.1.7 mise en correspondance d'adresses DL	[ISO/CEI 7498-1]
3.1.8 connexion DL	[ISO/CEI 7498-1]
3.1.9 point d'extrémité de connexion DL	[ISO/CEI 7498-1]
3.1.10 identifiant de point d'extrémité de connexion DL	[ISO/CEI 7498-1]
3.1.11 émission en mode connexion DL	[ISO/CEI 7498-1]
3.1.12 émission en mode sans connexion DL	[ISO/CEI 7498-1]
3.1.13 destination de données DL	[ISO/CEI 7498-1]
3.1.14 origine de données DL	[ISO/CEI 7498-1]
3.1.15 transmission en duplex DL	[ISO/CEI 7498-1]
3.1.16 fonctionnalité DL	[ISO/CEI 7498-1]
3.1.17 vue locale DL	[ISO/CEI 7498-3]
3.1.18 nom DL	[ISO/CEI 7498-3]
3.1.19 protocole DL	[ISO/CEI 7498-1]
3.1.20 identifiant de connexion de protocole DL	[ISO/CEI 7498-1]
3.1.21 information de contrôle de protocole DL	[ISO/CEI 7498-1]
3.1.22 unité de données de protocole DL	[ISO/CEI 7498-1]
3.1.23 identifiant de version de protocole DL	[ISO/CEI 7498-1]
3.1.24 relais DL	[ISO/CEI 7498-1]
3.1.25 identifiant de connexion de service DL	[ISO/CEI 7498-1]
3.1.26 unité de données de service DL	[ISO/CEI 7498-1]
3.1.27 transmission simplex DL	[ISO/CEI 7498-1]
3.1.28 sous-système DL	[ISO/CEI 7498-1]
3.1.29 données utilisateur DL	[ISO/CEI 7498-1]
3.1.30 contrôle de flux	[ISO/CEI 7498-1]
3.1.31 gestion de couche	[ISO/CEI 7498-1]
3.1.32 multiplexage	[ISO/CEI 7498-3]
3.1.33 autorité de nommage (adressage)	[ISO/CEI 7498-3]
3.1.34 domaine de nommage (adressage)	[ISO/CEI 7498-3]
3.1.35 sous-domaine de nommage (adressage)	[ISO/CEI 7498-3]

3.1.36 (N)-entité	[ISO/CEI 7498-1]
entité DL	
entité Ph	
3.1.37 (N)- unité de données d'interface	[ISO/CEI 7498-1]
unité de données de service DL (N=2)	
unité de données d'interface Ph (N=1)	
3.1.38 (N)-couche	[ISO/CEI 7498-1]
couche DL (N=2)	
couche Ph (N=1)	
3.1.39 (N)-service	[ISO/CEI 7498-1]
service DL (N=2)	
service Ph (N=1)	
3.1.40 (N)-point d'accès de service	[ISO/CEI 7498-1]
point d'accès de service DL (N=2)	
point d'accès de service Ph (N=1)	
3.1.41 (N)-adresse de point d'accès de service	[ISO/CEI 7498-1]
adresse de point d'accès de service DL (N=2)	
adresse de point d'accès de service Ph (N=1)	
3.1.42 entités homologues	[ISO/CEI 7498-1]
3.1.43 informations de commande d'interface Ph	[ISO/CEI 7498-1]
3.1.44 données d'interface Ph	[ISO/CEI 7498-1]
3.1.45 nom de primitive	[ISO/CEI 7498-3]
3.1.46 réassembler	[ISO/CEI 7498-1]
3.1.47 recombinaison	[ISO/CEI 7498-1]
3.1.48 réinitialiser	[ISO/CEI 7498-1]
3.1.49 adresse DL répondante	[ISO/CEI 7498-3]
3.1.50 acheminement	[ISO/CEI 7498-1]
3.1.51 segmenter	[ISO/CEI 7498-1]
3.1.52 séquencer	[ISO/CEI 7498-1]
3.1.53 subdiviser	[ISO/CEI 7498-1]
3.1.54 nom synonyme	[ISO/CEI 7498-3]
3.1.55 gestion systèmes	[ISO/CEI 7498-1]

3.2 Termes, définitions et conventions des services

La présente norme utilise également les termes suivants définis dans l'ISO/CEI 10731 dans la mesure où ils s'appliquent à la couche liaison de données:

- 3.2.1 **accepteur**
- 3.2.2 **service asymétrique**
- 3.2.3 **confirmation (primitive);
requestor.deliver (remise au demandeur) (primitive)**
- 3.2.4 **remise (primitive)**
- 3.2.5 **fonctionnalité confirmée DL**
- 3.2.6 **fonctionnalité DL**
- 3.2.7 **vue locale DL**
- 3.2.8 **fonctionnalité obligatoire DL**
- 3.2.9 **fonctionnalité non confirmée DL**
- 3.2.10 **machine protocolaire DL**
- 3.2.11 **fonctionnalité lancée par le fournisseur DL**
- 3.2.12 **fonctionnalité facultative du fournisseur DL**
- 3.2.13 **primitive de service DL;
primitive**
- 3.2.14 **fournisseur de service DL**
- 3.2.15 **utilisateur de service DL**
- 3.2.16 **fonctionnalité facultative d'utilisateur DLS**
- 3.2.17 **indication (primitive d');
acceptor.deliver (soumission à l'accepteur) (primitive)**
- 3.2.18 **multi-homologue**
- 3.2.19 **demande (primitive);
requestor.submit (soumission au demandeur) (primitive)**
- 3.2.20 **demandeur**
- 3.2.21 **réponse (primitive);
requestor.submit (soumission à l'accepteur) (primitive)**
- 3.2.22 **soumission (primitive)**
- 3.2.23 **service symétrique**

3.3 Termes et définitions communs

NOTE De nombreuses définitions sont communes à plusieurs types de protocoles; elles ne sont pas nécessairement utilisées par tous les types de protocoles.

3.3.1

segment, liaison, liaison locale DL

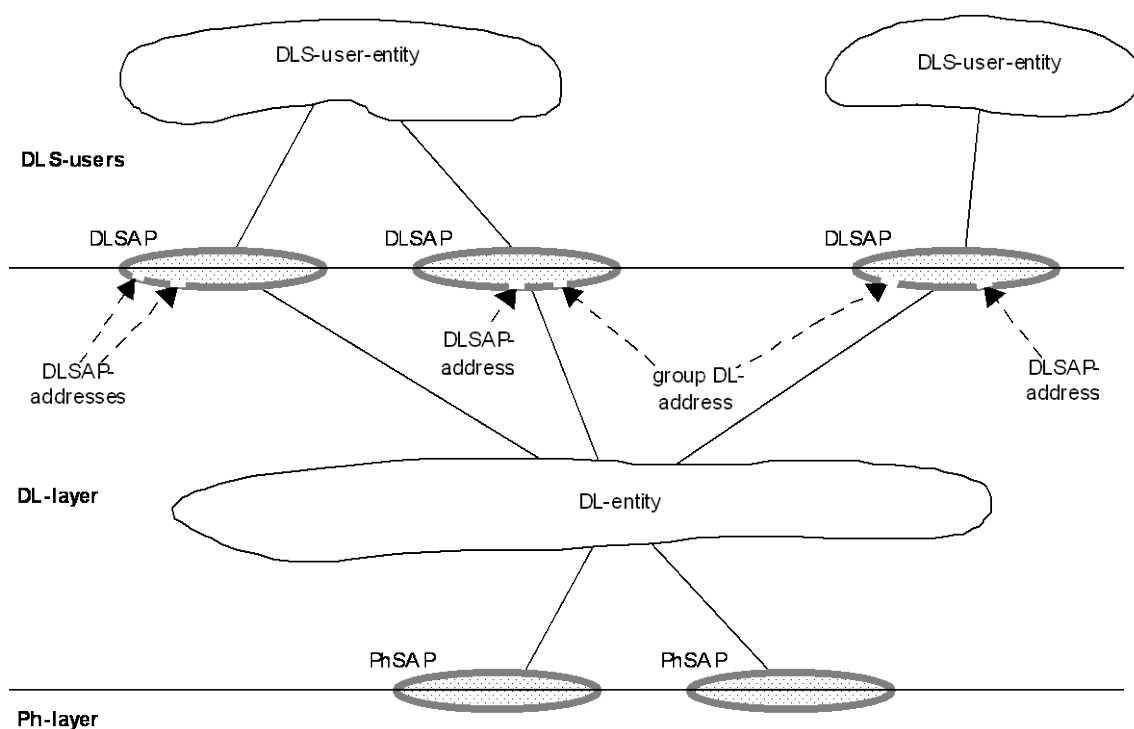
sous-réseau DL unique dans lequel l'une des DLE connectées peut communiquer directement sans l'intervention d'un relais DL, à chaque fois que toutes les DLE participant à une instance de communication sont simultanément attentives au sous-réseau DL pendant la/les période(s) de tentative de communication

3.3.2

point d'accès de service de liaison de données (DLSAP, *data-link service access point*)

point distinctif où des services DL sont fournis par une DLE simple à une entité simple de couche supérieure

NOTE Cette définition, tirée de l'ISO/CEI 7498-1, est ici réitérée pour faciliter la compréhension de l'importante distinction qui existe entre les DLSAP et leurs adresses DL. (Voir la Figure 1).



Légende

Anglais	Français
DLS-user-entity	Entité utilisateur DLS
DLS users	Utilisateurs DLS
DLSAP-addresses	Adresses DLSAP
DLSAP-address	Adresse DLSAP
Group-DL-address	Adresse DL de groupe
DLSAP-address	Adresse DLSAP
DL-layer	Couche DL
DL-entity	Entité DL
Ph-layer	Couche physique

NOTE 1 Les DLSAP et les points d'accès de service de couche physique (PhSAP) sont décrits comme des ovales qui recouvrent la frontière entre deux couches adjacentes.

NOTE 2 Les adresses DL sont décrites comme désignant de petits intervalles (points d'accès) dans la partie DLL d'un DLSAP.

NOTE 3 Une DLE simple peut avoir plusieurs adresses DLSAP et adresses DL de groupe, associées à un DLSAP simple.

Figure 1 – Relations entre DLSAP, adresses DLSAP et adresses DL de groupe

3.3.3

adresse DL(SAP)

il s'agit soit d'une adresse DLSAP individuelle, désignant un simple DLSAP d'un simple utilisateur (utilisateur DLS) d'un service de liaison de données (DLS), soit d'une adresse DL de groupe désignant potentiellement plusieurs DLSAP appartenant chacun à un simple utilisateur DLS

NOTE Cette terminologie a été choisie parce que l'ISO/CEI 7498-3 ne permet pas d'utiliser le terme adresse DLSAP pour désigner plusieurs DLSAP simples au niveau d'un utilisateur DLS simple.

3.3.4

adresse DLSAP (individuelle)

adresse DL qui désigne un seul DLSAP au sein de la liaison étendue

NOTE Une entité DL simple peut avoir plusieurs adresses DLSAP, associées à un DLSAP simple.

3.3.5

adresse de point d'extrémité de connexion de liaison de données (adresse DLCEP, *data-link connection endpoint*)

adresse DL qui désigne:

- a) soit un point d'extrémité de connexion DL homologue;
- b) soit un point d'extrémité de connexion DL éditeur multi-homologue, et implicitement l'ensemble correspondant de points d'extrémité de connexion DL d'abonnés, où chaque point d'extrémité de connexion DL existe au sein d'un DLSAP distinct et est associé à l'adresse DLSAP distincte correspondante

3.3.6

erreur de séquence de contrôle de trame (FCS, *frame check sequence*)

erreur qui apparaît lorsque la valeur calculée de la séquence de contrôle de trame, après réception de tous les octets dans une unité de données de protocole de liaison de données (DLPDU, *data link protocol data unit*) ne correspond pas à la valeur résiduelle attendue

3.3.7

trame

synonyme de DLPDU

3.3.8

gestion du réseau

fonctions et services de gestion qui réalisent l'initialisation, la configuration et le traitement des erreurs du réseau

3.3.9

protocole

convention relative aux formats des données, aux séquences temporelles et à la correction des erreurs pour l'échange de données dans des systèmes de communication

3.3.10

utilisateur DLS de réception

utilisateur de services DL qui agit comme destinataire de données utilisateur DLS

NOTE Un utilisateur de services DL peut être simultanément un utilisateur DLS d'émission et de réception.

3.3.11

utilisateur DLS d'émission

utilisateur de services DL qui agit comme origine de données utilisateur DLS

3.3.12

dispositif

DLE simple telle qu'elle apparaît sur une liaison locale

3.3.13

identifiant d'entité DL

adresse qui désigne la DLE (simple) associée à un dispositif simple sur une liaison locale spécifique

3.3.14**identifiant unique de dispositif**

identifiant unique, sur 8 octets, pour reconnaître un dispositif de Type 21 sur un réseau donné. Cet ID est une combinaison de l'adresse MAC sur 6 octets de l'ISO/CEI 8802-3:2000 et de l'adresse DL sur 2 octets

3.3.15**anneau**

réseau actif dans lequel chaque nœud est connecté en série à deux autres dispositifs

NOTE Un anneau peut également être désigné par le terme "boucle".

3.3.16**topologie linéaire**

topologie dans laquelle les dispositifs sont connectés ensemble en série, deux dispositifs étant chacun connecté à un seul autre dispositif, et tous les autres dispositifs étant chacun connecté à deux autres dispositifs, qui sont par exemple connectés sur une ligne

3.3.17**port R**

dans un dispositif de communication, port qui fait partie d'une structure annulaire

3.3.18**temps réel**

aptitude d'un système à fournir un résultat requis dans un temps limité

3.3.19**communication en temps réel**

transfert de données en temps réel

3.3.20**ethernet temps réel (RTE, *Real-time Ethernet*)**

réseau qui repose sur l'ISO/CEI 8802-3:2000 et qui inclut une communication en temps réel

NOTE 1 D'autres types de communication peuvent être pris en charge, à condition de ne pas compromettre la communication en temps réel.

NOTE 2 Cette définition se fonde, de manière nullement limitative, sur l'ISO/CEI 8802-3:2000. Elle pourrait être applicable à d'autres spécifications IEEE802, par exemple l'IEEE802.11.

3.3.21**dispositif d'extrémité RTE**

dispositif qui a au moins un port RTE actif

3.3.22**port RTE**

point de sous-couche de contrôle d'accès au support physique (MAC) utilisé pour relier un RTE au réseau local (LAN, *local area network*)

NOTE Cette définition est dérivée de celle d'un port de pont dans l'ISO/CEI 10038: 1993, comme appliquée aux ponts MAC locaux.

3.3.23**réseau commuté**

réseau comprenant également des commutateurs

NOTE Le terme réseau commuté signifie que le réseau est fondé sur les spécifications IEEE802.1D et IEEE802.1Q avec des ponts MAC et des opérations prioritaires.

3.3.24 liaison

trajet de transmission entre deux nœuds adjacents [tirée de l'ISO/CEI 11801]

3.4 Symboles et abréviations

3.4.1 Symboles et abréviations communs

DL	Liaison de données (utilisée comme préfixe ou adjectif)
DLC	Connexion de liaison de données (<i>data link connection</i>)
DLCEP	Point d'extrémité de connexion de liaison de données
DLE	Entité de Liaison de Données (l'instance active locale de la DLL)
DLL	Couche Liaison de données (<i>data link layer</i>)
DLPDU	Unité de données de protocole de liaison de données
DLPM	Machine protocolaire de liaison de données (<i>data link protocol machine</i>)
DLM	Gestion de liaison de données (<i>data link management</i>)
DLME	Entité de gestion de liaison de données (l'instance active locale de la DLM) (<i>data link management entity</i>)
DLMS	Service de gestion de liaison de données (<i>data link management service</i>)
DLS	Service de liaison de données (<i>data link service</i>)
DLSAP	Point d'accès de service de liaison de données
DLSDU	Unité de données de service de liaison de données (<i>data link service-data-unit</i>)
FIFO	Premier entré, premier sorti (méthode de mise en file d'attente) (<i>first-in, first-out</i>)
NMT	Gestion du réseau (<i>network management</i>)
OSI	Interconnexion de systèmes ouverts (<i>Open Systems Interconnection</i>)
Ph	Couche physique (utilisée comme préfixe)
PHY	Émetteur-récepteur d'interface physique
PhL	Couche physique (<i>physical layer</i>)
RTE	Ethernet temps réel
CEI	Commission Electrotechnique Internationale
IP	Protocole Internet (<i>Internet Protocol</i>) (voir RFC 791)
ISO	Organisation Internationale de Normalisation (<i>International Organization for Standardization</i>)
MAC	Contrôle d'accès au support physique (<i>media access control</i>)
NRT	Non temps réel (<i>non-real-time</i>)
PDU	Unité de données de protocole (<i>protocol data unit</i>)
SAP	Point d'accès de service (<i>service access point</i>)
RT	Temps réel (<i>real-time</i>)
TCP	Protocole de contrôle de transmission (<i>Transmission Control Protocol</i>) (voir RFC 793)
UDP	Protocole de datagramme d'utilisateur (<i>User Datagram Protocol</i>) (voir RFC 768)

3.4.2 Type 21: Symboles et abréviations supplémentaires

EFR	Rétablissement extrêmement rapide (<i>extremely fast recovery</i>)
GD	Dispositif générique (<i>general device</i>)

LNM	Gestionnaire de réseau linéaire (<i>line network manager</i>)
PO	Mise sous tension (<i>power on</i>)
PnP	Prêt à l'emploi (<i>plug and play</i>)
RNM	Gestionnaire de réseau annulaire (<i>ring network manager</i>)
RNMP	Gestionnaire de réseau annulaire primaire (<i>primary ring network manager</i>)
RNMS	Gestionnaire de réseau annulaire secondaire (<i>secondary ring network manager</i>)
RNAC	Auto-configuration de réseau annulaire (<i>ring network auto configuration</i>)
UID	Identifiant unique de dispositif (<i>device unique identification</i>)
NMIB de Type 21	Base d'informations de gestion de réseau (<i>network management information base</i>) de Type 21

4 Vue d'ensemble du protocole de liaison de données

4.1 Généralités

Le Type 21 étend l'utilisation d'Ethernet conformément à l'ISO/CEI 8802-3:2000, en lui conférant des mécanismes de transfert de données avec des demandes de temporisation prévisibles, caractéristiques d'une automatisation de haute performance. Il ne modifie pas les principes de base de la norme Ethernet ISO/CEI 8802-3:2000, mais l'étend vers l'Ethernet en temps réel (RTE). Il est ainsi possible de continuer à utiliser du matériel, des composants d'infrastructure ou des équipements d'essai et de mesure normalisés de l'Ethernet, tels que des analyseurs de réseau.

4.2 Vue d'ensemble du contrôle d'accès au support physique

Un dispositif de Type 21 exige un commutateur intégré à deux ports (ports d'anneau) connecté à l'anneau. Un système de réseau de Type 21 est constitué de dispositifs de commutation Ethernet, en duplex intégral, sans collision, en réseau annulaire ou linéaire. Le Type 21 garantit une transmission de données sans collision entre deux dispositifs reliés par une connexion Ethernet en duplex intégral. Ainsi, la couche liaison de données de Type 21 assure une transmission de données fiable, transparente et exempte de collision entre utilisateurs DLS.

Une connexion de Type 21 permet des communications de données exemptes de collision et en duplex intégral entre deux dispositifs. Par conséquent, un dispositif de Type 21 peut transmettre une trame à tout moment, sans aucune restriction en termes de droits d'accès au support. La couche de liaison de données de Type 21 ne limite pas la méthode de planification de l'utilisation des ressources de la liaison de données, mais chaque dispositif peut être programmé individuellement par l'applicatif, de sorte que le Type 21 puisse être appliqué en toute flexibilité dans diverses applications.

Une trame de Type 21 est remise au dispositif de destination selon l'une des deux méthodes suivantes:

- Une trame est lancée par un dispositif d'origine et directement transmise au dispositif de destination voisin;
- Une trame est lancée par un dispositif d'origine et acheminée vers le dispositif de destination par des dispositifs intermédiaires.

Lorsqu'une trame est acheminée par le dispositif intermédiaire, la procédure d'acheminement de trame est traitée par le commutateur matériel interne afin de réduire la durée de traitement; ceci n'affecte pas les performances de la DLL de Type 21.

4.3 Service pris en charge à partir de la couche physique

La présente norme décrit le service de couche physique (PhS) pris en charge ainsi que les contraintes liées à son utilisation par la DLE. Le service de couche physique est supposé fournir les primitives de service suivantes, spécifiées dans l'Article 2 de l'ISO/CEI 8802-3:2000.

Les primitives de PhS prises en charge sont MA-DATA.request (demande de DONNEES D'ACTIVITÉ DU SUPPORT) et MA-DATA.indication (indication de DONNEES D'ACTIVITÉ DU SUPPORT).

La relation temporelle entre primitives est présentée en Figure 2.

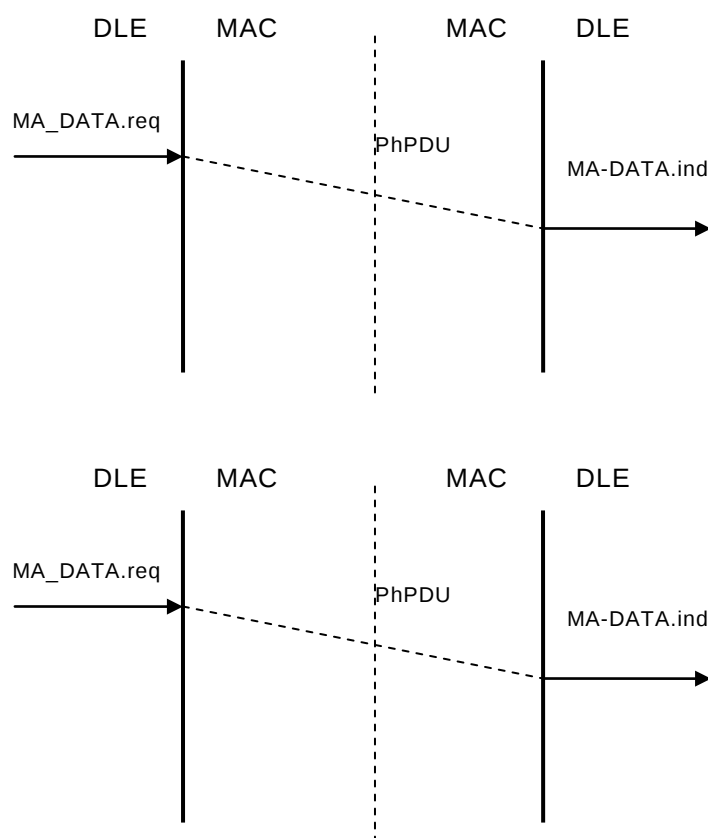


Figure 2 – Interaction des primitives de PhS avec une DLE

La primitive de demande MA-DATA définit le transfert de données d'une entité cliente MAC vers une entité homologue unique ou des entités homologues multiples dans le cas d'adresses de groupe.

La primitive d'indication MA-DATA définit le transfert de données depuis l'entité de sous-couche MAC (par l'intermédiaire de la couche de contrôle MAC facultative, si elle est mise en œuvre) vers l'entité ou les entités clientes MAC dans le cas d'adresses de groupe.

4.4 Architecture DLL

4.4.1 Généralités

La DLL de Type 21 fournit un service de transfert de données de niveau supérieur, fiable et efficace, en utilisant une topologie annulaire ou linéaire, en duplex intégral, sans aucun schéma spécifique de contrôle d'accès. La DLL de Type 21 est modélisée comme une

combinaison de composantes de contrôle de la machine protocolaire de liaison de données (DLPM) et de l'interface de gestion DLL.

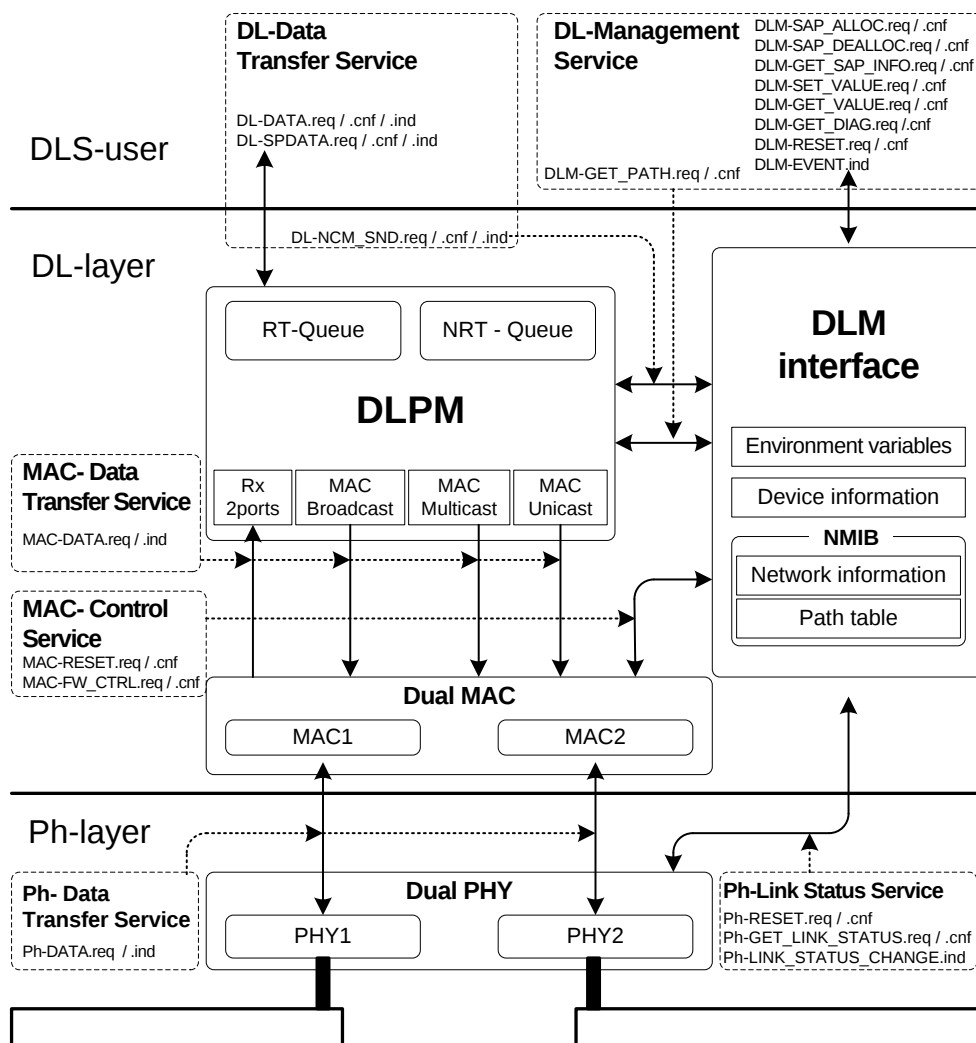
La DLPM transmet l'unité de données de service de liaison de données (DLSDU) générée par l'utilisateur DLS local, au port R approprié, conformément à la stratégie du service de liaison de données. La DLPM examine également la trame reçue et remet la DLPDU reçue à l'utilisateur DLS approprié.

L'interface de gestion DLL assure les fonctions de gestion DLL pour le maintien de la base d'informations de gestion de réseau (NMIB). La NMIB contient des informations concernant le dispositif local, des informations concernant le réseau ainsi que la table des trajets. La DLL est constituée des éléments énumérés dans le Tableau 1.

Tableau 1 – Composants de la DLL

Composants	Description
Machine protocolaire DL (DLPM)	Elle transmet la DLSDU reçue de l'utilisateur DLS local, via la file d'attente temps réel (file d'attente RT) ou non temps réel (file d'attente NRT). Elle examine la trame reçue et remet la DLPDU à l'utilisateur DLS approprié.
Interface de gestion DLL (Interface DLM)	Elle détient des variables de gestion de station qui appartiennent à la DLL et gère les modifications synchronisées des paramètres de la liaison.

La disposition interne de ces composants, ainsi que leurs interfaces, sont illustrées en Figure 3. Les têtes de flèche représentent la direction principale du flux de données et du contrôle.



Légende

Anglais	Français
DLS-user	Utilisateur DLS
DL-Data transfer service	Service de transfert de données DL
DL-management service	Service de gestion DL
DL-layer	Couche DL
RT-queue	File d'attente RT
NRT-queue	File d'attente NRT
DLM interface	Interface DLM
MAC-data transfer service	Service de transfert de données MAC
Rx 2 ports	2 ports Rx
MAC broadcast	Diffusion MAC
MAC multicast	Multidiffusion MAC
MAC unicast	Monodiffusion MAC
Environment variables	Variables d'environnement
Device information	Informations de dispositif
Network information	Informations de réseau
Path table	Table des trajets
MAC control service	Service de contrôle MAC

Anglais	Français
Dual MAC	MAC double
Ph-Layer	Couche physique
Ph-data transfer service	Service de transfert de données Ph
Dual PHY	PHY double
Ph-link status service	Service d'état de liaison Ph

Figure 3 – Architecture de la couche de liaison de données

4.4.2 Fonction de prise en charge de l'interface de gestion DLL (DLM)

La DLM est l'une des principales caractéristiques de Type 21 qui permet des fonctions telles que "prêt à l'emploi" (Plug and Play – PnP), "auto-configuration du réseau" (NAC) et "rétablissement extrêmement rapide" (EFR). La DLM maintient spontanément et partage des informations de dispositif local ainsi que des informations concernant le réseau avec chaque dispositif présent sur le réseau.

- Gestion NMIB de Type 21

L'interface DLM assure le maintien de la NMIB qui contient les informations de dispositif local et indique l'état physique et logique du dispositif, les informations de réseau qui indiquent l'état courant de la configuration du réseau et la table des trajets qui indique les informations de trajet aux autres dispositifs ainsi que leurs profils.

- Fonction "prêt à l'emploi" (PnP)

Lorsqu'un dispositif rejoint le réseau existant, il est automatiquement détecté et configuré sans nécessité de réglage manuel des paramètres, de façon à ce que le dispositif puisse directement communiquer avec les autres dispositifs du réseau.

- Auto-configuration du réseau (NAC)

Le Type 21 prend en charge des topologies réseau annulaires et linéaires. Lorsque le réseau change de topologie (d'annulaire à linéaire ou de linéaire à annulaire), cette modification est automatiquement détectée et diffusée à chaque dispositif sur le réseau, et ainsi les informations du réseau et les entrées de la table des trajets sont automatiquement mises à jour.

- Rétablissement extrêmement rapide (EFR)

En cas de changement de topologie annulaire en topologie linéaire, les informations du réseau et les informations relatives au trajet sont automatiquement mises à jour dans les 10 ms.

- Services de diagnostic système

L'interface DLM assure également un service de diagnostic système.

4.5 Type de données

4.5.1 Généralités

La présente norme décrit les types de données fondamentales de Type 21. Les types de données décrits dans la présente norme sont uniquement les références normatives utilisées pour représenter les formats des données de Type 21. La présente norme ne traite pas des questions de mise en œuvre.

4.5.2 Données booléennes

Les données fondamentales de type BOOLÉEN peuvent prendre des valeurs "VRAI" ou "FAUX". Les valeurs sont représentées comme des séquences binaires de longueur 1. La valeur "VRAI" est représentée par un 1 et la valeur "FAUX" est représentée par un 0.

4.5.3 Entier non signé

Les données fondamentales de type UNSIGNEDn (entier NON SIGNÉ) ont des valeurs d'entiers non négatifs. La plage de valeurs va de 0,..., à 2^{n-1} . Les données sont représentées comme des séquences binaires de longueur n .

$$b = b_0 \sim b_{n-1}$$

Il est attribué à la séquence binaire une valeur définie par

$$\text{UNSIGNEDn} = b_0 2^0 + b_1 2^1 + \dots + b_{n-1} 2^{n-1}$$

La séquence binaire commence à droite, par l'octet de poids faible, comme présenté dans le Tableau 2.

Tableau 2 – Type de données UNSIGNEDn

Numéro d'octet	1	2	3	4	5	6	7	8
UNSIGNED8	$b_0..b_7$							
UNSIGNED16	$b_0..b_7$	$b_8..b_{15}$						
UNSIGNED24	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$					
UNSIGNED32	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$				
UNSIGNED40	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$			
UNSIGNED48	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$		
UNSIGNED56	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$	$b_{48}..b_{55}$	
UNSIGNED64	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$	$b_{48}..b_{55}$	$b_{56}..b_{63}$

4.5.4 Entier signé

Les données fondamentales de type INTEGERn ont des valeurs entières. La plage de valeurs va de -2^{n-1} à 2^{n-1} . Les données sont représentées comme des séquences binaires de longueur n . Il est attribué à la séquence binaire une valeur définie par

$$b = b_0 \sim b_{n-1}$$

$$\text{INTEGERn} = b_0 2^0 + b_1 2^1 + \dots + b_{n-2} 2^{n-2}, \text{ lorsque } b_{n-1} \text{ est 0, et}$$

$$\text{INTEGERn} = -\text{Complément}(b) - 1, \text{ lorsque } b_{n-1} \text{ est 1,}$$

La séquence binaire commence à droite, par l'octet de poids faible, comme présenté dans le Tableau 3.

Tableau 3 – Type de données INTEGERn

Numéro d'octet	1	2	3	4	5	6	7	8
INTEGER8	$b_0..b_7$							
INTEGER16	$b_0..b_7$	$b_8..b_{15}$						
INTEGER24	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$					
INTEGER32	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$				
INTEGER40	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$			
INTEGER48	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$		
INTEGER56	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$	$b_{48}..b_{55}$	
INTEGER64	$b_0..b_7$	$b_8..b_{15}$	$b_{16}..b_{23}$	$b_{24}..b_{31}$	$b_{32}..b_{39}$	$b_{40}..b_{47}$	$b_{48}..b_{55}$	$b_{56}..b_{63}$

4.5.5 Chaîne d'octets

Le type de données OCTET_STRINGn (CHAÎNE D'OCTETS_n) est défini ci-dessous. n représente la longueur en octets de la chaîne d'octets.

ARRAY[n] of UNSIGNED8 OCTET_STRINGn - MATRICE[n] de CHAÎNE D'OCTETS_n NONSIGNES8

4.5.6 Chaîne de caractères visible

Le type de données VISIBLE_STRINGn (CHAÎNE DE CARACTÈRES VISIBLE_n) est défini ci-dessous. Les valeurs admissibles de données de type VISIBLE_CHAR sont 0x00 et la plage admissible est de 0x20 à 0x7E. Les données sont interprétées comme des caractères codés sur 7 bits. n indique la longueur d'octet de la chaîne de caractères visible.

UNSIGNED8 VISIBLE_CHAR

ARRAY[n] of UNSIGNED8 OCTET_STRINGn - MATRICE[n] de CHAÎNE D'OCTETS_n NONSIGNES8

Il n'y a pas de 0x00 nécessaire pour terminer la chaîne de caractères.

4.5.7 Heure du Jour

Le type de données TIMEOFDAY (HEURE DU JOUR) est défini ci-dessous. TIMEOFDAY est constitué de champs de comptage de la date UNSIGNED16 (NON SIGNÉ16) et de champs en millisecondes UNSIGNED32 (NON SIGNÉ32).

TIMEOFDAY

UNSIGNED32, en millisecondes: comptage du temps à partir de l'instant 00:00 en millisecondes.

UNSIGNED16, le comptage de la date: le nombre de jour depuis le 01.01.1984.

4.6 Paramètres et variables locaux

4.6.1 Généralités

La présente norme utilise les paramètres de demande d'utilisateur DLS P(...) et des variables locales V(...) comme moyen de clarifier les effets de certaines actions et les conditions dans lesquelles ces actions sont valides. La présente norme utilise également des temporisateurs locaux T(...) comme moyen de surveillance des actions du fournisseur DLS distribué et comme moyen de garantir une réponse à la DLE locale en l'absence de ces actions. La norme utilise des compteurs locaux C(...) pour exécuter des fonctions de mesure du débit. Elle utilise également des files d'attente locales Q(...) comme moyen d'ordonnancer certaines activités, de clarifier les effets de certaines actions et de clarifier les conditions dans lesquelles ces activités sont valides.

Sauf spécification contraire, au moment de leur création ou de l'activation de la DLE:

- a) Toutes les variables doivent être initialisées à leur valeur par défaut ou à leur valeur minimale admissible, s'il n'est spécifié aucune valeur par défaut;
- b) Tous les compteurs doivent être initialisés à zéro;
- c) Tous les temporisateurs doivent être initialisés à l'état inactif.

La DLM peut modifier les valeurs des variables de configuration.

Les paramètres et variables locaux comprennent des paramètres de configuration de la DLE, des informations concernant le dispositif local et des variables NMIB. L'état opérationnel de la DLE est stocké dans les paramètres de configuration de la DLE et les paramètres de configuration de la DLE sont configurés localement ou à distance en fonction de l'application. Les informations de la liaison de données locales, telles que l'identifiant de l'entité DL et l'état de chaque port R, sont stockées dans les informations relatives au dispositif local. La NMIB inclut des informations concernant le réseau ainsi que la table des trajets. La topologie du réseau et les variables relatives au réseau sont stockées dans les informations de réseau; le profil du dispositif et les informations de trajet des autres dispositifs sont résumés dans la table des trajets.

4.6.2 Paramètres de configuration DLE

4.6.2.1 Généralités

Ces paramètres sont nécessaires pour configurer le fonctionnement de la DLE locale; ils sont gérés par la DLM. Chaque dispositif sur le même segment de réseau dispose des mêmes paramètres de configuration DLE. Le service DLM-GET_VALUE (OBTENIR VALEUR DLM) et le service DLM-SET_VALUE (ÉTABLIR VALEUR DLM) sont utilisés pour lire ou écrire des paramètres de configuration DLE. Le Tableau 4 présente la liste des paramètres de configuration DLE.

Tableau 4 – Paramètres de configuration DLE

Paramètre	Type de Données	Valeur par défaut	Description
Identifiant d'entité DL max	UNSIGNED16	255	Identifiant d'entité DL maximum 256 à 65 535: Réservé
Stratégie de planification DLPM	UNSIGNED8	0	0: Premier entré, Premier sorti (FIFO) 1: Priorité fixe 2: à 255: Réservé

4.6.2.2 P(MAX_ADDR): Identifiant d'entité DL maximum

Cette variable contient l'adresse maximale de dispositif; elle est établie par la DLM. La plage de cette variable va de 1 à 255. La valeur par défaut de cette variable est 255. Cette variable

indique également le nombre maximal d'entrées dans la table des trajets de la NMIB. Les valeurs dans la plage 256 à 65 535 sont réservées.

4.6.2.3 P(DLPMSP): Stratégie de planification de la machine protocolaire de liaison de données

Cette variable contient la stratégie de planification de la DLPM locale, qui indique comment se servir des DLSDU concurrentes dans la file d'attente RT. La file d'attente NRT n'est pas planifiée par la DLPMSP. La DLPMSP peut prendre l'une des valeurs suivantes:

- a) 0: Premier entré, Premier sorti (FIFO). La première trame reçue est servie en premier. Dans ce cas, la priorité du message dans la DLSDU est ignorée;
- b) 1: Priorité fixe. La DLSDU prioritaire est servie en premier. Si plusieurs DLSDU sont empilées avec la même priorité, la trame reçue en premier est servie en premier;
- c) 2 à 255: Réservés

4.6.3 Files d'attente pour la prise en charge du transfert de données

4.6.3.1 Généralités

Lorsqu'un utilisateur DLS génère une DLSDU à envoyer par la DLPM, la DLSDU est encapsulée par la DLPDU. La DLPDU est ensuite stockée dans la file d'attente RT ou dans la file d'attente NRT selon son type d'application et de service. Le Tableau 5 présente la liste des files d'attente pour le transfert de données de Type 21.

Tableau 5 – Files d'attente pour la prise en charge du transfert de données

Paramètre	Type de Données	Valeur par défaut	Description
File d'attente RT	—	—	File d'attente de transmission pour le stockage des données en temps réel
File d'attente NRT	—	—	File d'attente de transmission pour le stockage des données non temps réel

4.6.3.2 Q(RTQ): File d'attente RT

La file d'attente RT est utilisée pour stocker les DLPDU en temps réel générées par l'utilisateur DLS à transmettre par la DLPM. La taille de la file d'attente RT n'est pas limitée dans la présente norme et on considère qu'il s'agit d'un détail lié à la mise en œuvre locale. La file d'attente RT est planifiée par la DLPM.

4.6.3.3 Q(NRTQ): File d'attente NRT

La file d'attente NRT stocke des DLPDU non temps réel générées par un utilisateur DLS. La taille de la file d'attente NRT n'est pas limitée dans la présente norme et on considère qu'il s'agit d'un détail lié à la mise en œuvre locale. Les messages dans la file d'attente RT sont transmis en premier. Les messages dans la file d'attente NRT sont transmis uniquement lorsque la file d'attente RT est vide.

4.6.4 Variables de prise en charge de la gestion SAP

4.6.4.1 Généralités

L'affectation et la désaffectation des points d'accès de service de liaison de données(DLSAP) sont gérées par la DLM. Lorsqu'une trame est reçue, la DLPM examine le point d'accès de service de destination (DSAP) dans la DLSDU. Si le DSAP est déjà attribué à l'utilisateur DLS, la DLM renvoie l'ID de l'utilisateur DLS approprié qui équivaut à l'adresse DSAP reçue, et la DLPM remet la DLSDU reçue à l'utilisateur DLS. Si le point d'accès de service (SAP) n'est pas attribué et qu'il n'est trouvé aucun ID d'utilisateur DLS approprié dans la DLM, la DLSDU reçue est ignorée par la DLPM. Par conséquent, pour recevoir une DLSDU d'un quelconque

utilisateur DLS homologue, un utilisateur DLS doit d'abord obtenir une attribution de SAP au moyen du service DLM-SAP_ALLOC (ATTRIBUTION DE POINT SAP DE GESTION DLM). Une fois le SAP attribué à un utilisateur DLS, il est utilisé pour envoyer et recevoir des données jusqu'à ce que le SAP ne soit plus attribué et qu'il soit renvoyé à la DLM. Le SAP dont l'attribution a été retirée peut à nouveau être utilisé après ré-attribution. L'adresse SAP et l'ID d'utilisateur DLS correspondant sont stockés ensemble et conservés par la DLM. Le nombre maximal d'éléments de gestion de SAP est de 65 535 mais la méthode d'octroi et de retrait d'attribution de l'adresse SAP n'est pas limitée dans la présente norme. Le Tableau 6 présente la liste des variables de gestion de SAP.

Tableau 6 – Variables de prise en charge de la gestion SAP

Paramètre	Type de Données	Valeur par défaut	Description
SAP	UNSIGNED16	—	Point d'accès de service
ID d'utilisateur DLS	UNSIGNED32	—	Identification numérique de l'utilisateur DLS auquel est attribué le SAP

4.6.4.2 V(SAP): SAP

Cette variable contient le SAP. La plage de valeurs de cette variable est de 0 à 65 535.

4.6.4.3 V(DLS_USER_ID): ID d'utilisateur DLS

Cette variable contient l'identification numérique sur 4 octets de l'utilisateur DLS auquel le SAP est attribué.

4.6.5 Variables de prise en charge de la gestion des informations du dispositif local

4.6.5.1 Généralités

Pour maintenir la topologie du réseau, chaque dispositif gère une base de données de dispositif qui comprend les informations de dispositif local et les informations concernant les autres dispositifs. Le Tableau 7 présente la liste des variables de gestion des informations de dispositif.

Tableau 7 – Variables de prise en charge de la gestion des informations du dispositif

Paramètre	Type de Données	Valeur par défaut	Description
Identifiant d'entité DL	UNSIGNED16	INVALID_ADDR	Identifiant d'entité DL locale
Indicateurs de dispositif	UNSIGNED16	0	Indicateurs de dispositif local
Type de dispositif	UNSIGNED16	0	Type de dispositif local
Nombre de sauts	UNSIGNED16	0	Nombre de sauts
UID de dispositif	UNSIGNED64	INVALID_UID	Identifiant unique de dispositif local
UID de dispositif pour le Port R1	UNSIGNED64	INVALID_UID	Identifiant unique du dispositif connecté par l'intermédiaire du Port R1
UID de dispositif pour le Port R2	UNSIGNED64	INVALID_UID	Identifiant unique du dispositif connecté par l'intermédiaire du Port R2
Adresse MAC	UNSIGNED48	0	Adresse MAC de dispositif local
Réservé0	UNSIGNED16	0	Réservé
Informations de port	UNSIGNED16	0	Informations du port de dispositif local
Etat du dispositif	UNSIGNED8	0	Etat DLM
Version du protocole	UNSIGNED8	0	Version de protocole du dispositif local
Description de dispositif	VISIBLE_STRING[16]	" "	Chaîne de caractères de description du dispositif
Réservé1	UNSIGNED32	0	Réservé

4.6.5.2 V(DL_ADDR): Identifiant d'entité DL

Cette variable contient l'identifiant d'entité DL qui désigne l'entité DL (unique) associée à un dispositif unique sur une liaison locale spécifique et dont la valeur est limitée dans une plage de 0 à 255. L'identifiant d'entité DL peut être fourni par des réglages matériels (par exemple un commutateur rotatif) ou par logiciel. L'identifiant d'entité DL est défini dans le Tableau 8.

Tableau 8 – Identifiant d'entité DL

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	Lire	0 à 255: Identifiant d'entité DL 256 à 65 535: réservés

4.6.5.3 V(DEV_FLAG): Indicateur de dispositif

Cette variable contient les indicateurs d'événements qui ont eu lieu dans un dispositif local. Lorsque l'indicateur de collision de l'identifiant d'entité DL locale est établi, l'événement EVENT_THIS_ADDR_COLLISION est généré par la DLM, et l'indicateur de collision de l'identifiant d'entité DL locale est retiré. Lorsque l'indicateur de changement d'état DLM est établi, la DLM génère EVENT_DEV_STATE_CHG et l'indicateur de changement d'état de la DLM est retiré. Les indicateurs de dispositif et d'événement de Type 21 sont énumérés dans le Tableau 9.

Tableau 9 – Indicateurs de dispositif

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	Lire	0x00: normal 0x01: collision 0x02: modifié 0x03: collision et modifié Tous les autres sont réservés

4.6.5.4 V(DLM_STATE): Etat DLM

Cette variable contient l'état DLM. L'état DLM est défini comme indiqué dans le Tableau 10.

Tableau 10 – Etat DLM

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED8	Lire/Ecrire	Lire	0: INVALID_DLM_STATE 1: état autonome (SA) 2: état de gestionnaire de réseau linéaire (LNM) 3: état de dispositif générique (GD) 4: état de gestionnaire de réseau annulaire primaire (RNMP) 5: état de gestionnaire de réseau annulaire secondaire (RNMS) 6 à 255: Réserve

4.6.5.5 V(DEV_UID): UID de dispositif

Cette variable contient l'identifiant unique sur 8 octets d'un dispositif de Type 21 sur un réseau donné; elle est constituée d'une combinaison de l'adresse MAC sur 6 octets de l'ISO/CEI 8802-3:2000 et de l'identifiant de l'entité DL sur 2 octets. L'UID de dispositif est défini comme indiqué dans le Tableau 11.

Tableau 11 – Identifiant unique de dispositif

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED64	Bit 0 à 15	Lire/Ecrire	Lire	0 à 255: Identifiant d'entité DL 256 à 65 535: Réserve
	Bit 16 à 63	Lire/Ecrire	Lire	Adresse MAC de l'ISO/CEI 8802:3:2000

4.6.5.6 V(DEV_UID_RP1): UID de dispositif pour le Port R1

Cette variable contient l'UID du dispositif qui est relié par l'intermédiaire du Port R1. L'UID de dispositif pour le Port R1 est défini comme indiqué dans le Tableau 12.

Tableau 12 – Identifiant unique de dispositif connecté au Port R1

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED64	Bit 0 à 15	Lire/Ecrire	Lire	0 à 255: Identifiant d'entité DL 256 à 65 535: Réservé
	Bit 16 à 63	Lire/Ecrire	Lire	Adresse MAC de l'ISO/CEI 8802:3:2000

4.6.5.7 V(DEV_UID_RP2): UID de dispositif pour le Port R2

Cette variable contient l'UID du dispositif qui est relié par l'intermédiaire du Port R2. L'UID de dispositif pour le Port R2 est défini comme indiqué dans le Tableau 13.

Tableau 13 – Identifiant unique de dispositif connecté au Port R2

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED64	Bit 0 à 15	Lire/Ecrire	Lire	0 à 255: Identifiant d'entité DL 256 à 65 535: Réservé
	Bit 16 à 63	Lire/Ecrire	Lire	Adresse MAC de l'ISO/CEI 8802:3:2000

4.6.5.8 V(MAC_ADDR): adresse MAC

Cette variable contient l'adresse MAC Ethernet sur 6 octets du dispositif local, comme défini dans l'ISO/CEI 8802-3:2000. Sachant qu'un dispositif de Type 21 dispose de deux ports Ethernet, il convient que les deux adresses MAC soient identiques. L'adresse MAC est définie comme indiqué dans le Tableau 14.

Tableau 14 – Adresse MAC

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED48	Lire/Ecrire	Lire	Adresse MAC de l'ISO/CEI 8802:3:2000

4.6.5.9 V(PORT_INFO): Informations de port

Cette variable contient les informations de port pour chaque port R. Elle est définie comme indiqué dans le Tableau 15.

Tableau 15 – Informations de port

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Bit 0	Lire/Ecrire	Lire	Liaison Port R1 hors-service
	Bit 1	Lire/Ecrire	Lire	Message NCM_FAMILY_RES reçu du port R1
	Bit 2	Lire/Ecrire	Lire	Attente du message NCM_ADV_THIS reçu du port R1
	Bit 3	Lire/Ecrire	Lire	Attente du message NCM_MEDIA_LINKED du port R1
	Bit 4	Lire/Ecrire	Lire	Confirmation de l'état du port R1
	Bit 5 à 7	Lire/Ecrire	Lire	Réservé
	Bit 8	Lire/Ecrire	Lire	Liaison Port R2 hors-service
	Bit 9	Lire/Ecrire	Lire	Message NCM_FAMILY_RES reçu du port R2
	Bit 10	Lire/Ecrire	Lire	Attente du message NCM_ADV_THIS du port R2
	Bit 11	Lire/Ecrire	Lire	Attente du message NCM_MEDIA_LINKED du port R2
	Bit 12	Lire/Ecrire	Lire	Confirmation de l'état du port R2
	Bit 13 à 15	Lire/Ecrire	Lire	Réservé

4.6.5.10 V(PROTOCOL_VER): Version du protocole

Cette variable contient la version de protocole du dispositif local. Elle est définie comme indiqué dans le Tableau 16.

Tableau 16 – Version de protocole

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED8	Bit 0 à 1	Lire	Lire	0x00: version majeure 1 0x01: version majeure 2 0x02: version majeure 3 0x03: version majeure 4
	Bit 2 à 4	Lire	Lire	0x00: version mineure 0 0x01: version mineure 1 0x02: version mineure 2 0x03: version mineure 3 0x04: version mineure 4 0x05: version mineure 5 0x06: version mineure 6 0x07: version mineure 7
	Bit 5 à 7	—	—	Réservé

4.6.5.11 V(DEV_TYPE): Type de dispositif

Cette variable contient le type de dispositif local qui représente la fonction générale du dispositif. La valeur de cette variable est définie comme indiqué dans le Tableau 17.

Tableau 17 – Type de dispositif

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Bit 0 à 7	Lire	Lire/Ecrire	0 à 255: type de dispositif générique 0: type de dispositif non valide 1: automate programmable (PLC) 2: dispositif de commande de mouvement 3: interface homme-machine(IHM) 4: ordinateur personnel industriel 5: onduleur 6: dispositif d'E/S simple 7 à 255: réservés
	Bit 8 à 15	Lire	Lire/Ecrire	0 à 255: type de dispositif spécifique à l'application

4.6.5.12 V(DEV_DESC): Description de dispositif

Cette variable contient une description du dispositif local. Sa longueur maximale est de 16 octets. Elle est définie comme indiqué dans le Tableau 18.

Tableau 18 – Description de dispositif

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
VISIBLE_STRING[16]	—	Lire/Ecrire	Toute chaîne de caractères définie par un utilisateur DLS établie en utilisant le service d'établissement de configuration de DLL

4.6.5.13 V(HOP_CNT): Nombre de sauts

Cette variable contient l'indication du nombre de dispositifs entre deux dispositifs. Lorsque la DLM reçoit NCM_ADV_THIS ou DLPDU NCM_LA, elle sauvegarde la valeur du nombre de sauts reçue dans cette variable, puis incrémente de 1 le nombre de sauts dans la trame reçue et transmet la trame par l'autre port R. De cette manière, chaque dispositif construit sa propre table des trajets avec un nombre de sauts pour le port R1 et un nombre de sauts pour le port R2. Cette variable est définie comme indiqué dans le Tableau 19.

Tableau 19 – Nombre de sauts

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	—	0 à 255: Nombre de sauts 256 à 65 535: réservés

4.6.6 Variables et compteur de prise en charge de la gestion des informations du réseau

4.6.6.1 Généralités

Les informations du réseau sont gérées automatiquement par la DLM. Les variables et les compteurs d'informations du réseau sont résumés dans le Tableau 20.

Tableau 20 – Variables de prise en charge des informations de gestion du réseau

Paramètre	Type de Données	Valeur par défaut	Description
Topologie	UNSIGNED8	NET_TPG_SA	Topologie du réseau
Nombre de collisions	UNSIGNED8	0	Compteur de collisions d'identifiant d'entité DL entre deux dispositifs distants
Nombre de dispositifs	UNSIGNED16	1	Compteur de dispositif pour le segment de réseau
Nombre de changements de topologie	UNSIGNED16	0	Compteur de changement de topologie du réseau
Indicateurs de réseau	UNSIGNED16	0	Indicateurs d'événements réseau
Heure du dernier changement de topologie	TIMEOFDAY	0	Date et heure du dernier changement de topologie du réseau.
Réservé	UNSIGNED16	0	Réservé
UID de dispositif RNMP	UNSIGNED64	INVALID_UID	UID du dispositif gestionnaire RNMP
UID de dispositif RNMS	UNSIGNED64	INVALID_UID	UID du dispositif gestionnaire RNMS
UID du dispositif LNM pour le port R1	UNSIGNED64	INVALID_UID	UID du dispositif LNM dans le sens du port R1
UID du dispositif LNM pour le port R2	UNSIGNED64	INVALID_UID	UID du dispositif LNM dans le sens du port R2

4.6.6.2 V(TPG): Topologie

Cette variable contient le type de topologie du réseau. La valeur de cette variable est définie comme indiqué dans le Tableau 21.

Tableau 21 – Topologie

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED8	Lire/Ecrire	Lire	Topologie du réseau 0x01: autonome 0x02: topologie linéaire 0x03: Topologie annulaire

4.6.6.3 C(COLL_CNT): Nombre de collisions

Cette variable contient le nombre de collisions d'identifiant d'entité DL pour des dispositifs distants. Cette valeur est incrémentée par la DLM lorsqu'il est détecté une collision d'identifiants d'entité DL distante et elle est décrémentée lorsque la collision est éliminée. Cette variable est définie dans le Tableau 22.

Tableau 22 – Nombre de collisions

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED8	Lire/Ecrire	Lire	0 à 255: Nombre de collisions d'identifiant d'entité DL distante

4.6.6.4 C(DEV_CNT): Nombre de dispositifs

Cette variable contient le nombre total de dispositifs sur le réseau, jusqu'à un maximum de 256. Elle est définie comme indiqué dans le Tableau 23.

Tableau 23 – Nombre de dispositifs

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	Lire	0: non utilisé 1 à 256: nombre de dispositifs 257 à 65 535: réservés

4.6.6.5 C(TPG_CHG_CNT): Nombre de changements de topologie

Cette variable contient le nombre de changements de topologie. La valeur est incrémentée par la DLM lorsque la topologie du réseau change d'annulaire en linéaire ou de linéaire en annulaire. Elle est définie comme indiqué dans le Tableau 24.

Tableau 24 – Nombre de changements de topologie

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	Lire	0 à 65 535: Nombre de changements de topologie

4.6.6.6 V(TPG_CHG_TIME): Heure du dernier changement de topologie

Cette variable contient la date et l'heure du dernier changement de topologie du réseau. Elle est définie comme indiqué dans le Tableau 25.

Tableau 25 – Heure du dernier changement de topologie

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
TIMEOFDAY	Lire/Ecrire	Lire	La date et l'heure du dernier changement de topologie du réseau.

4.6.6.7 V(UID_RNMP): UID de dispositif RNMP

Cette variable contient l'UID du dispositif sélectionné comme étant le gestionnaire RNMP du réseau. Elle est définie comme indiqué dans le Tableau 26.

Tableau 26 – UID de dispositif RNMP

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED64	Bit 0 à 15	Lire/Ecrire	Lire	0 à 255: Identifiant d'entité DL 256 à 65 535: Réserve
	Bit 16 à 63	Lire/Ecrire	Lire	Adresse MAC de l'ISO/CEI 8802:3:2000

4.6.6.8 V(UID_RNMS): UID de dispositif RNMS

Cette variable contient l'UID du dispositif sélectionné comme étant le gestionnaire RNMS du réseau. Elle est définie comme indiqué dans le Tableau 27.

Tableau 27 – UID de dispositif RNMS

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED64	Bit 0 à 15	Lire/Ecrire	Lire	0 à 255: Identifiant d'entité DL 256 à 65 535: Réserve
	Bit 16 à 63	Lire/Ecrire	Lire	Adresse MAC de l'ISO/CEI 8802:3:2000

4.6.6.9 V(UID_LNM_RP1): UID du dispositif LNM pour le port R1

Dans un réseau linéaire de Type 21, les deux dispositifs d'extrémité sont automatiquement choisis comme étant les gestionnaires LNM. Cette variable contient l'UID du dispositif choisi comme étant le gestionnaire LNM dans le sens du Port R1. Elle est définie comme indiqué dans le Tableau 28.

Tableau 28 – UID du dispositif LNM pour le port R1

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED64	Bit 0 à 15	Lire/Ecrire	Lire	0 à 255: Identifiant d'entité DL 256 à 65 535: Réserve
	Bit 16 à 63	Lire/Ecrire	Lire	Adresse MAC de l'ISO/CEI 8802:3:2000

4.6.6.10 V(UID_LNM_RP2): UID du dispositif LNM pour le port R2

Dans un réseau linéaire de Type 21, les deux dispositifs d'extrémité sont automatiquement choisis comme étant les gestionnaires LNM. Cette variable contient l'UID du dispositif choisi comme étant le gestionnaire LNM dans le sens du Port R2. Elle est définie comme indiqué dans le Tableau 29.

Tableau 29 – UID du dispositif LNM pour le port R2

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED64	Bit 0 à 15	Lire/Ecrire	Lire	0 à 255: Identifiant d'entité DL 256 à 65 535: Réservé
	Bit 16 à 63	Lire/Ecrire	Lire	Adresse MAC de l'ISO/CEI 8802:3:2000

4.6.6.11 V(NET_FLAG): Indicateurs de réseau

Cette variable contient les indicateurs d'événements pour informer l'utilisateur DLMS d'événements du réseau. Lorsque l'un des bits de cette variable est établi, la DLM génère une primitive de service d'indication d'événement DLM pour informer l'utilisateur DLMS de l'événement. Une fois le service d'événement DLM achevé avec succès, le bit désigné dans cette variable est remis à zéro. Les indicateurs du réseau sont définis dans le Tableau 30.

Tableau 30 – Indicateurs de réseau

Type de Données	Position	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Bit 0	Lire/Ecrire	Lire	Etat de changement de topologie du réseau 0x00: Normal 0x01: La topologie du réseau a changé
	Bit 1	Lire/Ecrire	Lire	Etat de collisions d'identifiant d'entité DL du réseau 0x00: Normal 0x01: Collision d'identifiant d'entité DL du réseau détectée
	Bit 2	Lire/Ecrire	Lire	Etat d'Entrée de dispositif 0x00: Normal 0x01: Un nouveau dispositif a rejoint le réseau
	Bit 3	Lire/Ecrire	Lire	Etat d'Entrée de dispositif 0x00: Normal 0x01: Un dispositif a quitté le réseau
	Bit 4 à 63	—	—	Réservé

Les champs binaires des indicateurs de réseau sont définis comme suit:

a) Etat de changement de topologie du réseau;

Ce bit est mis sur "VRAI" par la DLM lorsqu'elle détecte une modification de la topologie du réseau. Ce bit est remis sur "FAUX" lorsque l'utilisateur DLMS a été avisé au moyen du service EVENT_NET_TPG_CHG (EVENEMENT DE CHANGEMENT DE TOPOLOGIE DU RÉSEAU).

b) Etat de collision d'identifiant d'entité DL du réseau

Ce bit est mis sur "VRAI" par la DLM lorsqu'elle détecte une collision d'identifiant d'entité DL sur le réseau. Ce bit est remis sur "FAUX" lorsque la collision d'identifiant d'entité DL est éliminée.

c) Etat d'Entrée de dispositifs;

Ce bit est mis sur "VRAI" par la DLM lorsqu'elle détecte qu'un nouveau dispositif a rejoint le réseau. Ce bit est remis sur "FAUX" lorsque l'utilisateur DLMS a été avisé au moyen du service EVENT_IN_DEVICE (EVENEMENT D'ENTRÉE DE DISPOSITIF).

d) Etat de Sortie de dispositif

Ce bit est mis sur "VRAI" par la DLM lorsqu'elle détecte qu'un dispositif a quitté le réseau. Ce bit est remis sur "FAUX" lorsque l'utilisateur DLMS a été avisé au moyen du service EVENT_OUT_DEVICE (EVENEMENT DE SORTIE DE DISPOSITIF).

4.6.7 Variables et compteur de prise en charge de la gestion des informations de trajets de dispositifs

4.6.7.1 Généralités

La table des trajets est gérée par la DLM; elle est constituée d'éléments concernant les autres dispositifs de Type 21 présents sur le réseau. Les variables et compteurs de ces éléments sont définis dans le Tableau 31.

**Tableau 31 – Variables et compteur de prise en charge
de la gestion des informations de trajets**

Paramètre	Type de Données	Valeur par défaut	Description
Identifiant d'entité DL	UNSIGNED16	INVALID_ADDR	Identifiant d'entité DL
Nombre de sauts pour le Port R1	UNSIGNED16	INVALID_HOP_CNT	Nombre de sauts dans le sens du Port R1
Nombre de sauts pour le Port R2	UNSIGNED16	INVALID_HOP_CNT	Nombre de sauts dans le sens du Port R2
Port R préférentiel	UNSIGNED8	INVALID_R_PORT	Port R, vers le dispositif homologue, ayant la plus petite valeur de nombre de sauts.
Port R de destination	UNSIGNED8	INVALID_R_PORT	Port R sélectionné pour l'envoi d'une trame vers l'identifiant de l'entité DL de destination.
Etat du dispositif	UNSIGNED8	0	Etat DLM du dispositif homologue
Adresse MAC	UNSIGNED48	0	Adresse MAC de l'ISO/CEI 8802:3:2000 du dispositif homologue
Informations de port	UNSIGNED16	0	Informations de Port R local de dispositif homologue
Version du protocole	UNSIGNED8	0	Version de protocole du dispositif homologue
Type de dispositif	UNSIGNED16	0	Type de dispositif d'application du dispositif homologue
Description de dispositif	VISIBLE_STRING[16]	" "	Description du dispositif homologue
UID de dispositif	UNSIGNED64	INVALID_UID	UID de dispositif du dispositif homologue
UID de dispositif pour le Port R1	UNSIGNED64	INVALID_UID	UID de dispositif du dispositif connecté par le Port R1 du dispositif homologue
UID de dispositif pour le Port R2	UNSIGNED64	INVALID_UID	UID de dispositif du dispositif connecté par le Port R2 du dispositif homologue
Nombre d'entrées sur le réseau	UNSIGNED16	0	Le nombre de fois où le dispositif homologue a rejoint le réseau.
Horodatage de l'entrée sur le réseau	TIMEOFDAY	0	La date et l'heure de la dernière fois où le dispositif homologue a rejoint le réseau.
Nombre de sorties du réseau	UNSIGNED16	0	Le nombre de fois où le dispositif homologue a été déconnecté du réseau.
Horodatage de la sortie du réseau	TIMEOFDAY	0	La date et l'heure de déconnexion du dispositif homologue pour la dernière fois

4.6.7.2 V(path-DL_ADDR): Identifiant d'entité DL

Voir 4.6.5.2.

4.6.7.3 C(path-HOP_CNT_RP1): Nombre de sauts pour le Port R1

Cette variable indique le nombre d'acheminements de trames pour l'envoi d'une trame depuis le dispositif local jusqu'au dispositif homologue par l'intermédiaire du Port R1. Elle est définie comme indiqué dans le Tableau 32.

Tableau 32 – Nombre de sauts dans le sens du Port R1

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	Lire	0 à 255: Nombre de sauts pour le Port R1 256 à 65 535: Réserve

4.6.7.4 C(path-HOP_CNT_RP2): Nombre de sauts pour le Port R2

Cette variable indique le nombre d'acheminements de trames pour l'envoi d'une trame depuis le dispositif local jusqu'au dispositif homologue par l'intermédiaire du Port R2. Elle est définie comme indiqué dans le Tableau 33.

Tableau 33 – Nombre de sauts dans le sens du Port R2

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	Lire	0 à 255: Nombre de sauts pour le Port R2 256 à 65 535: Réserve

4.6.7.5 V(path-PREFER_RP): Port R préférentiel

Cette variable contient le Port R préférentiel pour l'envoi d'une trame depuis le dispositif local jusqu'au dispositif homologue, qu'il s'agisse d'un RNMP ou d'un RNMS. Cette variable est déterminée comme étant la plus petite valeur de nombre de sauts pour le dispositif homologue. Si le nombre de sauts du Port R1 et le nombre de sauts du Port R2 ont la même valeur, le Port R1 est sélectionné comme Port R préférentiel. Cette variable est définie comme indiqué dans le Tableau 34.

Tableau 34 – Port R préférentiel

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED8	Lire/Ecrire	Lire	0: Invalide 1: Port R1 2: Port R2 3 à 255: Réserve

4.6.7.6 V(path-DST_RP): Port R de destination

Cette variable contient le Port R de destination pour l'envoi d'une trame du dispositif local au dispositif homologue. Dans un réseau linéaire, cette variable a la même valeur que le Port R préférentiel. Cependant, dans un réseau annulaire, cette variable est déterminée en fonction de RNMP et de RNMS car le trajet préférentiel peut être bloqué par le RNMP ou le RNMS. Dans ce cas, le Port R de destination est sélectionné comme étant l'autre Port R. Elle est définie comme indiqué dans le Tableau 35.

Tableau 35 – Port R de destination

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED8	Lire/Ecrire	Lire	0: Invalide 1: Port R1 2: Port R2 3 à 255: Réserve

4.6.7.7 V(path-DEV_STATE): Etat du dispositif

Voir 4.6.5.4.

4.6.7.8 V(path-MAC_ADDR): Adresse MAC

Voir 4.6.5.8.

4.6.7.9 V(path-PORT_INFO): Informations de port

Voir 4.6.5.9.

4.6.7.10 V(path-PROTOCOL_VER): Version du protocole

Voir 4.6.5.10.

4.6.7.11 V(path-DEV_TYPE): Type de dispositif

Voir 4.6.5.11.

4.6.7.12 V(path-DEV_DESC): Description de dispositif

Voir 4.6.5.12.

4.6.7.13 V(path-DEV_UID): UID de dispositif

Voir 4.6.5.5.

4.6.7.14 V(path-DEV_UID_RP1): UID de dispositif pour le Port R1

Voir 4.6.5.6.

4.6.7.15 V(path-DEV_UID_RP2): UID de dispositif pour le Port R2

Voir 4.6.5.7.

4.6.7.16 C(path-IN_NET_CNT): Nombre d'Entrées sur le réseau

Cette variable contient le nombre de fois où le dispositif homologue a rejoint le réseau. Lorsqu'un réseau linéaire est fusionné dans un réseau existant, les variables concernant les dispositifs qui ont nouvellement rejoint le réseau sont simultanément incrémentées. Cette variable est définie comme indiqué dans le Tableau 36.

Tableau 36 – Nombre d'Entrées sur le réseau

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	Lire	0 à 65 535: Le nombre de fois où le dispositif homologué a rejoint le réseau.

4.6.7.17 V(path-IN_NET_TIME): Horodatage de l'Entrée sur le réseau

Cette variable contient la date et l'heure de la dernière fois où le dispositif homologué a rejoint le réseau. Cette variable est définie comme indiqué dans le Tableau 37.

Tableau 37 – Horodatage de l'Entrée sur le réseau

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
TIMEOFDAY	Lire/Ecrire	Lire	La date et l'heure de la dernière fois où le dispositif homologué a rejoint le réseau.

4.6.7.18 C(path-OUT_NET_CNT): Nombre de Sorties du réseau

Cette variable contient le nombre de fois où le dispositif homologué a été déconnecté du réseau. Lorsqu'un dispositif ou un groupe de dispositifs est(sont) déconnecté(s) du réseau, les variables correspondantes sont simultanément incrémentées. Cette variable est définie comme indiqué dans le Tableau 38.

Tableau 38 – Nombre de sorties du réseau

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
UNSIGNED16	Lire/Ecrire	Lire	0 à 65 535: Le nombre de fois où le dispositif a été déconnecté du réseau.

4.6.7.19 V(path-OUT_NET_TIME): Horodatage de la Sortie du réseau

Cette variable contient la date et l'heure de la dernière fois où le dispositif a été déconnecté du réseau. Cette variable est définie comme indiqué dans le Tableau 39.

Tableau 39 – Horodatage de la Sortie du réseau

Type de Données	DLM de type d'accès	Utilisateur DLS de type d'accès	Valeur/Description
TIMEOFDAY	Lire/Ecrire	Lire	La date et l'heure de déconnexion du dispositif pour la dernière fois

4.6.8 Variables, compteurs, temporisateurs et files d'attente de prise en charge de la gestion des tables des trajets**4.6.8.1 Table des trajets**

La table des trajets est constituée d'éléments concernant les autres dispositifs sur le réseau. Elle est gérée par la DLM sous la forme d'une matrice tabulaire renseignée avec les informations de trajet de chaque dispositif (voir 4.6.7). La taille maximale de la table des trajets dépend de MAX_ADDR (voir 4.6.2.2) comme présenté ci-dessous:

Table des trajets: Matrice[n] d'informations de trajets de dispositif, $n = \text{MAX_ADDR} + 1$

5 Structure générale et codage

5.1 Vue d'ensemble

La DLL et ses procédures sont nécessaires pour fournir des services à l'utilisateur DLS, au moyen des services mis à disposition à partir de la couche physique. Le présent Article décrit la structure et les règles sémantiques de l'unité de données de protocole d'application de gestion (MAPDU), la DLPDU et la procédure communément utilisées dans la présente norme. Cette partie est identique et pleinement conforme à l'ISO/CEI 8802-3:2000.

NOTE Dans le présent Article, toute référence au bit k d'un octet est une référence au bit dont le poids dans un entier non signé d'un octet est de 2^k . Ceci est parfois appelé numérotation binaire "petit-boutiste".

5.2 Structure et codage de MAPDU

La sous-couche MAC locale utilise les primitives de service fournies par la sous-couche de service de couche physique (PLS) spécifiée à l'Article 2 de l'ISO/CEI 8802-3:2000. Ces primitives de service fournies par la sous-couche PLS sont obligatoires:

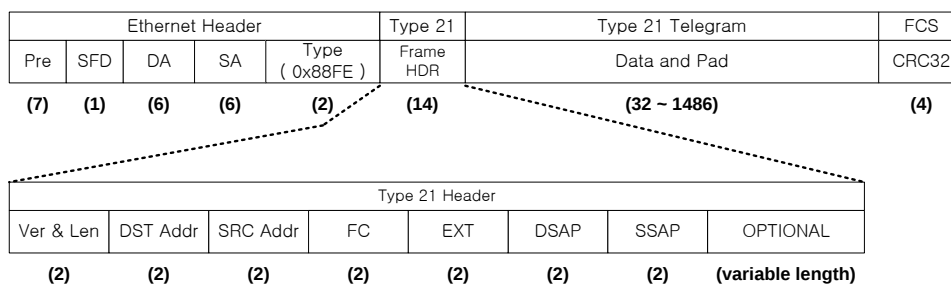
- a) MA-DATA request (demande de DONNEES D'ACTIVITE DU SUPPORT);
- b) MA-DATA indication (indication de DONNEES D'ACTIVITE DU SUPPORT);

5.3 Structure, codage et éléments de trame MAC commune

5.3.1 Structure de trame MAC

5.3.1.1 Format de trame MAC pour la DLPDU de Type 21

La DLPDU utilisée pour le Type 21 est encapsulée dans le champ de données d'une trame MAC comme spécifié dans l'Article 3 de l'ISO/CEI 8802-3:2000. La valeur du champ longueur/type est 88FE_H , autorisée et enregistrée comme numéro d'identification du protocole par l'organisme d'enregistrement de l'IEEE, pour identifier une trame de bus de terrain de Type 21. La structure de la DLPDU de Type 21 est illustrée en Figure 4.



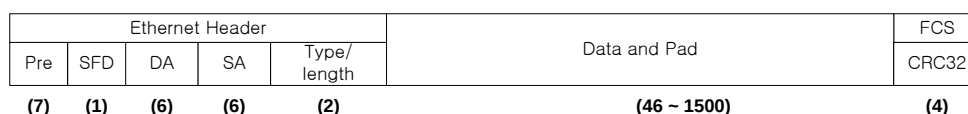
Légende

Anglais	Français
Ethernet header	En-tête Ethernet
Type 21 telegram	Télégramme de Type 21
Pre	Préambule
Frame HDR	En-tête de trame
Data and Pad	Données et remplissage
Type 21 header	En-tête de Type 21
Ver & Len	Version & Longueur
DST addr	adr. Destinataire
SRC addr	adr. ORIGINE
OPTIONAL	FACULTATIF
(variable length)	(longueur variable)

Figure 4 – Format de trame MAC commune pour la DLPDU de Type 21

5.3.1.2 Format de trame MAC pour DLPDU sporadique de bus de terrain de Type 21

Le format de trame MAC utilisé pour la transmission de données sporadiques de bus de terrain de Type 21 est identique au format de trame Ethernet V2.0 spécifié à l'Article 3 de l'ISO/CEI 8802-3:2000 " Structure de trame de contrôle d'accès au support physique"; et la valeur du champ longueur/type peut prendre toute valeur autre que 0x88FE. La Figure 5 illustre le format de trame pour une DLPDU sporadique de bus de terrain de Type 21.



Légende

Anglais	Français
Ethernet header	En-tête Ethernet
Pre	Préambule
Type / length	Type / longueur
Data and pad	Données et remplissage

Figure 5 – Format de trame MAC pour d'autres protocoles

5.3.2 Eléments de la trame MAC

5.3.2.1 Généralités

Les éléments de la trame MAC sont le préambule, le délimiteur de début de trame, l'adresse MAC de destination, l'adresse MAC d'origine, le code de longueur/type et la séquence de

contrôle de trame (FCS); tous ces éléments sont spécifiés à l'Article 3 de l'ISO/CEI 8802-3:2000.

5.3.2.2 Champ Préambule

Le préambule de la trame MAC est identique à celui décrit dans l'Article 3 de l'ISO/CEI 8802-3:2000. Il s'agit d'un champ de 7 octets utilisé pour permettre à la partie signalisation physique du circuit de parvenir à une synchronisation en régime établi avec la trame de réception. Le schéma du préambule est le suivant:

“10101010 10101010 10101010 10101010 10101010 10101010 10101010”

Les bits sont émis dans l'ordre, de gauche à droite. La nature du schéma est telle que, pour le codage Manchester, il apparaît comme une forme d'onde périodique sur le support, ce qui permet la synchronisation binaire. Il convient de noter que le préambule se termine par un “0”.

5.3.2.3 Délimiteur de début de trame

Le délimiteur de début de trame (SFD) est identique à celui décrit dans l'Article 3 de l'ISO/CEI 8802-3:2000. La séquence binaire du champ SFD est “10101011.” Elle suit immédiatement le schéma du préambule et indique le début d'une trame.

5.3.2.4 Champ Adresse MAC de destination

Le champ adresse MAC de destination est identique à celui décrit dans l'Article 3 de l'ISO/CEI 8802-3:2000. Il spécifie le(s) dispositif(s) au(x)quel(s) la trame est destinée et il peut s'agir d'une adresse individuelle ou multidestinataire (y compris la diffusion). L'adresse MAC de destination est configurée par la DLM sur l'identifiant d'entité DL correspondant. Le Type 21 définit également une adresse MAC spéciale, 00-E0-91-02-05-99 (NCM_MAC_ADDR) pour le partage des informations de gestion du réseau au moyen des services DLM. Chaque message reçu par l'intermédiaire de NCM_MAC_ADDR est remis à la DLM pour la mise à jour des informations de gestion du réseau. Le message n'est pas acheminé par la couche MAC mais examiné et acheminé par la DLM.

5.3.2.5 Champ Adresse MAC d'origine

Le champ adresse MAC d'origine est identique à celui décrit dans l'Article 3 de l'ISO/CEI 8802-3:2000. Ce champ indique le dispositif qui envoie la trame et n'est pas interprété par la DLE ou la sous-couche MAC CSMA/CD.

5.3.2.6 Champ longueur/type

Le champ longueur/type est identique à celui décrit dans l'Article 3 de l'ISO/CEI 8802-3:2000 “Structure de trame de contrôle d'accès au support physique.” Pour être identifiée comme une trame de Type 21, la valeur du champ longueur/type est établie à 0x88FE, autorisée et enregistrée comme étant le numéro d'identification du protocole pour RTE de Type 21 par l'organisme d'enregistrement de l'IEEE. Chaque trame ayant une valeur autre que 0x88FE est identique à la trame décrite dans l'Article 3 de l'ISO/CEI 8802-3:2000; elle est traitée comme une trame de données sporadiques de bus de terrain de Type 21.

5.3.2.7 Séquence de contrôle de trame

Le champ FCS est identique à celui décrit dans l'Article 3 de l'ISO/CEI 8802-3:2000.

5.3.3 Eléments de la DLPDU de Type 21

5.3.3.1 Version et Longueur

Ce champ contient la version de protocole et la longueur d'un télégramme ou d'un champ de données de Type 21. La Figure 6 illustre ce champ version et longueur. La version est

représentée par 2 bits lorsqu'il s'agit d'une version majeure et par 3 bits lorsqu'il s'agit d'une version mineure; la longueur est indiquée par 11 bits.

Version and Length															
Version					Length										
15	14	13	21	11	10	9	8	7	6	5	4	3	2	1	0

Légende

Anglais	Français
Version and length	Version et Longueur
Version	Version
Length	Longueur

Figure 6 – Champs version et longueur

Les différentes parties de ce champ et valeurs admissibles sont indiquées dans le Tableau 40.

Tableau 40 – Version et Longueur

Nom du champ		Position	Valeur/Description
Longueur		Bit 0 à 10	Longueur de trame y compris le champ FCS
Version	Mineure	Bit 11 à 13	Version mineure de protocole de Type 21 0x00: version mineure 0 0x01: version mineure 1 0x02: version mineure 2 0x03: version mineure 3 0x04: version mineure 4 0x05: version mineure 5 0x06: version mineure 6 0x07: version mineure 7
	Majeure	Bit 14 à 15	Version majeure de protocole de Type 21 0x00: version majeure 1 0x01: version majeure 2 0x02: version majeure 3 0x03: version majeure 4

5.3.3.2 DST_addr

5.3.3.2.1 Généralités

Ce champ indique l'identifiant de l'entité DL de destination du nœud auquel la trame est envoyée. La représentation de cette valeur est illustrée en Figure 7.

Destination DL-address															
DST_Addr															
15	14	13	21	11	10	9	8	7	6	5	4	3	2	1	0

Légende

Anglais	Français
Destination DL-address	Adresse DL de destination

Figure 7 – Champ DST_addr

Le champ individuel et ses valeurs admissibles sont décrits dans le Tableau 41.

Tableau 41 – Identifiant d'entité DL de destination

Nom du champ	Position	Valeur/Description
DST_addr	Bit 0 à 15	0xFFFF: Adresse de diffusion 0xFFFE: adresse de contrôle du réseau (C_NCM_ADDR) 0xFFFD-0xFFDE: adresse de multidiffusion définie par l'utilisateur 0xFFDD: adresse non valide 0x0100 à 0xFFDC: réservés 0x0000 à 0x00FF: identifiant ordinaire d'entité DL de Type 21

5.3.3.2.2 Adresse de diffusion

Si l'identifiant d'entité DL de destination est 0xFFFF, le champ adresse MAC de destination contient l'adresse MAC de l'ISO/CEI 8802-3:2000.

5.3.3.2.3 Adresse de contrôle du réseau

Si l'identifiant d'entité DL de destination est 0xFFFE (C_NCM_ADDR), le champ adresse MAC de destination contient C_NCM_MAC_ADDR. Cependant, les messages NCM_LINK_ACTV et NCM_ADV_THIS sont transmis en utilisant C_NCM_ADDR comme identifiant d'entité DL de destination.

NOTE L'utilisateur DLS ne peut pas accéder à C_NCM_ADDR.

5.3.3.2.4 Adresse de multidiffusion définie par l'utilisateur

Une adresse de multidiffusion définie par l'utilisateur est utilisée pour indiquer plusieurs destinataires. Cependant, l'adressage de multidiffusion défini par l'utilisateur n'est pas une fonctionnalité obligatoire dans la présente norme. Il est destiné à des systèmes d'application particuliers qui nécessitent une communication en multidiffusion. Par conséquent, l'adressage de multidiffusion défini par l'utilisateur ne comporte pas d'exigences d'interopérabilité entre dispositifs hétérogènes. L'identifiant d'entité DL de destination utilise la plage de 0xFFFD à 0xFFDE pour spécifier l'adresse de multidiffusion définie par l'utilisateur. Cependant, la présente norme ne spécifie pas la méthode d'utilisation de l'adresse de multidiffusion définie par l'utilisateur et considère qu'il s'agit d'une responsabilité locale. La présente norme ne limite pas l'utilisation des adresses de multidiffusion définies par l'utilisateur et il ne s'agit pas d'une fonctionnalité obligatoire.

5.3.3.3 SRC_addr

Ce champ indique l'identifiant de l'entité DL d'origine du nœud d'où la trame est générée. La représentation de cette valeur est illustrée en Figure 8.

Source DL-address															
SRC_Addr															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Légende

Anglais	Français
Source DL-address	Adresse DL d'origine

Figure 8 – Champ SRC_addr

Le champ individuel et ses valeurs admissibles sont décrits dans le Tableau 42.

Tableau 42 – Identifiant d'entité DL d'origine

Nom du champ	Position	Valeur/Description
SRC_addr	Bit 0 à 15	Identifiant d'entité DL d'origine

5.3.3.4 Contrôle de trame (FC)

5.3.3.4.1 Généralités

Le champ contrôle de trame indique les informations de contrôle de trame. La représentation de cette valeur est illustrée en Figure 9.

Frame Control															
VoE	RES	PRI		ToS				NCMT							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

key

VoE: Validation of Extension code

RES: Reserved

PRI : Priority

ToS: Type of Service

NCMT: Network control message type

Légende

Anglais	Français
Frame control	Contrôle de trame
Key	Légende
VoE: Validation extension code	VoE: validation du code d'extension
RES: Reserved	Res: réservé
PRI: Priority	PRI: priorité
ToS: Type of service	ToS: type de service
NCMT: Network control message type	NCMT: type de message de contrôle du réseau

Figure 9 – Champ Contrôle de trame

Le champ individuel et ses valeurs admissibles sont décrits dans le Tableau 43.

Tableau 43 – Contrôle de trame

Nom du champ	Position	Valeur/Description
Type de Message de contrôle du réseau (NCMT)	Bit 0 à 7	0x00: réservé 0x01: NCM_FAMILY_REQ 0x02: NCM_FAMILY_RES 0x03: NCM_MEDIA_LINKED 0x04: NCM_ADV_THIS 0x05: NCM_LINE_START 0x06: NCM_RING_START 0x07: NCM_ACK_RNMS 0x08: NCM_RETRY_RNMS 0x09 à 0xFF: réservé
Type de Service (ToS)	Bit 8 à 11	0x00: Message de contrôle du réseau (NCM) 0x01: demande de service non confirmée 0x02-0x0F: Réservés
Priorité (PRI)	Bit 12 à 13	0x00: priorité la plus basse ... 0x03: priorité la plus élevée
Réservé	Bit 14	Réservé
Validation du Code d'extension (VoE)	Bit 15	0x00: Code d'EXT non valide 0x01: Code d'EXT valide

5.3.3.4.2 Validation du Code d'extension (VoE)

Si la trame dispose du champ extension, VoE est mis sur "VRAI"; dans le cas contraire VoE est mis sur "FAUX".

5.3.3.4.3 Priorité

Ce champ indique la priorité de la trame. Ce champ contient la valeur du paramètre de priorité du message pour le service DL. La priorité la plus élevée est 0x03 et la plus basse est 0x00.

5.3.3.4.4 Type de Service (ToS)

Ce champ indique le type de service DL. Une valeur de 0x00 indique un message de contrôle du réseau entre DLM, et 0x01 indique une demande de service non confirmée entre utilisateurs DLS.

5.3.3.4.5 Type de Message de contrôle du réseau (NCMT)**5.3.3.4.5.1 Généralités**

Le champ NCMT indique le type de message de contrôle du réseau.

5.3.3.4.5.2 NCM_FAMILY_REQ

NCMT: 0x01

Ce message de contrôle du réseau est utilisé pour demander le dispositif nouvellement connecté par l'intermédiaire d'un Port R s'il s'agit d'un dispositif de Type 21. Le message de

contrôle du réseau est transmis par l'intermédiaire du Port R nouvellement activé. Ce message ne doit pas être acheminé vers l'autre port.

5.3.3.4.5.3 NCM_FAMILY_RES

NCMT: 0x02

Ce message de contrôle du réseau est utilisé pour confirmer si le destinataire est un dispositif de Type 21 lorsque ce destinataire reçoit le message NCM_FAMILY_REQ du dispositif nouvellement connecté. Ce message est transmis par l'intermédiaire du Port R utilisé pour recevoir le message NCM_FAMILY_REQ. Ce message ne doit pas être acheminé vers l'autre port.

5.3.3.4.5.4 NCM_MEDIA_LINKED

NCMT: 0x03

Ce message de contrôle du réseau est utilisé pour indiquer qu'une nouvelle liaison de Type 21 a été établie par l'intermédiaire du Port R. Ce message est transmis par l'intermédiaire du Port R nouvellement activé. L'identifiant d'entité DL de destination contient C_NCM_ADDR. Lorsque la DLM reçoit ce message, elle incrémente le nombre de sauts dans la trame et transmet cette trame par l'intermédiaire de l'autre Port R. Ce message est ignoré par la LNM ou par le dispositif qui a généré le message.

5.3.3.4.5.5 NCM_ADV_THIS

NCMT: 0x04

Ce message de contrôle du réseau est utilisé pour transmettre les informations concernant le dispositif local du destinataire lorsqu'il reçoit un message NCM_MEDIA_LINKED provenant d'un nouveau dispositif sur le réseau. Ce message est transmis par l'intermédiaire du Port R utilisé pour recevoir le message NCM_MEDIA_LINKED. L'identifiant d'entité DL de destination contient C_NCM_ADDR. Lorsque la DLM reçoit ce message, elle incrémente le nombre de sauts dans la trame et transmet cette trame par l'intermédiaire de l'autre Port R. Ce message est ignoré par la LNM ou par le dispositif qui a généré le message.

5.3.3.4.5.6 NCM_LINE_START

NCMT: 0x05

Ce message de contrôle du réseau est utilisé pour diffuser le fait que la topologie du réseau a été automatiquement configurée en réseau linéaire. Ce message est lancé par la DLM dont l'état est changé en LNM lorsque le réseau linéaire existant est divisé en deux réseaux linéaires ou lorsqu'une défaillance de liaison est détectée sur un réseau annulaire et qu'il est reconfiguré en réseau linéaire. Ce message est diffusé sur le réseau en utilisant l'adresse de diffusion.

5.3.3.4.5.7 NCM_RING_START

NCMT: 0x06

Ce message de contrôle du réseau est utilisé pour diffuser le fait que la topologie du réseau a été automatiquement configurée en réseau annulaire. Ce message est lancé et diffusé par l'intermédiaire de deux Ports R par la DLM dont l'état a été changé en RNMP.

5.3.3.4.5.8 NCM_ACK_RNMS

NCMT: 0x07

Ce message de contrôle du réseau est utilisé par le dispositif RNMS pour diffuser le fait qu'il a été sélectionné avec succès. Ce message est transmis du RNMS au RNMP.

5.3.3.4.5.9 NCM_RETRY_RNMS

NCMT: 0x08

Ce message de contrôle du réseau est utilisé pour demander une retransmission du message NCM_ACK_RNMS provenant du dispositif RNMS dans le cas où le dispositif RNMP ne l'a pas initialement reçu. Le message NCM_RETRY_RNMS est transmis du RNMP au RNMS.

5.3.3.5 Extension (EXT)

5.3.3.5.1 Généralités

Ce champ existe lorsque le bit VoE du champ de contrôle de trame est mis sur "VRAI". Le champ extension est représenté en Figure 10.

Extension Code															
G	Extension Type							Extension Length							
15	14	13	21	11	10	9	8	7	6	5	4	3	2	1	0

key
G: Group Mask

Légende

Anglais	Français
Extension code	Code d'extension
Extension type	Type d'extension
Extension length	Longueur d'extension
key	Légende
G: Group Mask	G: masque de groupe

Figure 10 – Champ Extension

Le champ individuel et ses valeurs admissibles sont décrits dans le Tableau 44.

Tableau 44 – Extension

Nom du champ	Position	Valeur/Description
Longueur d'extension	Bit 0 à 7	0 à 255: la longueur du champ Extension
Type d'extension	Bit 8 à 14	0: type d'extension non valide 1 à 127: réservé pour un futur usage
Activation de masque de groupe	Bit 15	0x0: Le masque de groupe est activé 0x1: Le masque de groupe est désactivé

5.3.3.5.2 Activation de masque de groupe

Activation de masque de groupe est un champ binaire qui permet de spécifier si la trame doit être acceptée ou non par le dispositif homologue lorsqu'elle est envoyée en diffusion ou en multidiffusion. Lorsque la valeur est mise sur "VRAI", le masque de groupe est activé dans le dispositif homologue qui reçoit la trame. Dans le cas contraire, le masque de groupe est

désactivé dans le dispositif homologué. Lorsque le masque de groupe est activé, les champs masque de groupe sont annexés au champ option (voir 5.3.3.8).

5.3.3.5.3 Type d'extension

Ce champ indique le type de champ d'extension. La valeur 0x00 indique un type d'extension non valide et les autres valeurs sont réservées pour un usage futur.

5.3.3.5.4 Longueur d'extension

Ce champ indique la longueur du champ extension. Lorsque l'activation du masque de groupe est mise sur "VRAI" et que le type d'extension est mis sur 0x00, la longueur d'extension spécifie la longueur du champ masque de groupe. Lorsque l'activation du masque de groupe est mise sur "FAUX" et que le type d'extension est mis sur une valeur autre que 0x00, la longueur d'extension spécifie la longueur du champ extension. Lorsque l'activation du masque de groupe est mise sur "VRAI" et que le type d'extension n'est pas mis sur 0x00, les deux premiers octets spécifient la longueur du champ masque de groupe et les deux octets suivants spécifient le type d'extension.

5.3.3.6 DSAP

Ce champ indique le SAP de la DLE à laquelle la DLPDU est envoyée. La plage de valeurs admissibles est de 0 à 65 535. La représentation du DSAP est illustrée en Figure 11 et décrite dans le Tableau 45.

Destination service access point															
DSAP															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Légende

Anglais	Français
Destination service access point	Point d'accès de service de destination

Figure 11 – Champ DSAP

Tableau 45 – Point d'accès de service de destination

Nom du champ	Position	Valeur/Description
DSAP	Bit 0 à 15	Point d'accès de service de la DLE de destination

5.3.3.7 SSAP

Ce champ indique le SAP de la DLE d'où est générée la DLPDU. La plage de valeurs admissibles est de 0 à 65 535. La représentation du DSAP est illustrée en Figure 12 et décrite dans le Tableau 46.

Source service access point															
SSAP															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Légende

Anglais	Français
Source service access point	Point d'accès de service d'origine

Figure 12 – Champ Point d'accès de service d'origine

Tableau 46 – Point d'accès de service d'origine

Nom du champ	Position	Valeur/Description
SSAP	Bit 0 à 15	Point d'accès de service de la DLE d'origine

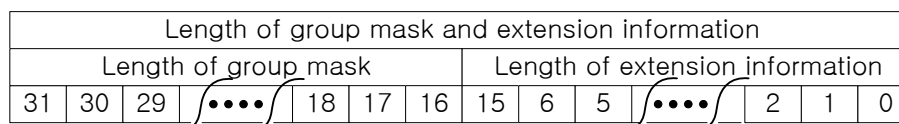
5.3.3.8 Option

5.3.3.8.1 Généralités

Ce champ indique le champ option lorsque la VoE (voir 5.3.3.4.2) est mise sur "VRAI". Le champ Option est utilisé pour indiquer des informations de masque de groupe ou d'autres informations supplémentaires. La longueur maximale du champ option est limitée à 256 octets.

5.3.3.8.2 Longueur des informations de masque de groupe et d'extension

Ce champ indique la longueur des informations masque de groupe et extension. Lorsque l'activation du masque de groupe est mise sur "VRAI" et que le type d'extension n'est pas mis sur 0x00, les deux premiers octets indiquent la longueur du champ masque de groupe et les deux octets suivants indiquent la longueur du champ type d'extension. Lorsque l'activation du masque de groupe est mise sur "FAUX" et que le type d'extension est mis sur 0x00, la longueur du champ d'informations de masque de groupe et d'extension est ignorée.



Légende

Anglais	Français
Length of group mask and extension information	Longueur des informations de masque de groupe et d'extension
Length of group mask	Longueur du masque de groupe
Length of extension information	Longueur des informations d'extension

Figure 13 – Longueur des informations de masque de groupe et d'extension

5.3.3.8.3 Masque de groupe

Ce champ utilise une séquence binaire pour indiquer le choix de réception d'un message. Lorsque l'activation de masque de groupe est mise sur "VRAI", le champ masque de groupe est annexé dans les unités de 4 octets, par exemple, 4, 8, 12 à 32 octets. Chaque bit indique le choix de réception de la trame pour l'identifiant d'entité DL correspondant. Un 1 signifie "VRAI" pour réception de la trame et 0 signifie "FAUX". Le premier bit indique l'option de réception de la trame pour l'identifiant d'entité DL le plus élevé. La Figure 14 présente l'ordre des séquences binaires du champ masque de groupe lorsque sa longueur est mise sur 255. Lorsque le type d'extension est mis sur 0x00, le champ masque de groupe est annexé au champ option. Dans le cas contraire, le champ masque de groupe est annexé après la longueur des champs d'informations de masque de groupe et d'extension.



Légende

Anglais	Français
Group Mask	Masque de groupe
Up to 255	jusqu'à 255

Figure 14 – Champ option de masque de groupe

5.3.3.8.4 Informations d'extension

Ce champ est réservé pour une extension future. Ce champ contient 4 octets d'informations d'extension.

5.3.3.9 Données et remplissage

Ce champ indique le champ de données reçu d'un utilisateur DLS.

5.4 Ordre d'émission des bits

L'ordre d'émission des bits est identique à celui décrit dans l'Article 3 de l'ISO/CEI 8802-3:2000. Chaque octet de la DLPDU, à l'exception de FCS, est transmis avec le bit d'ordre inférieur en premier.

5.5 DLPDU non valide

Une DLPDU non valide doit être définie comme étant une DLPDU qui satisfait au moins à l'une des conditions suivantes; cette définition est quasiment identique à l'Article 3 de l'ISO/CEI 8802-3:2000:

- la longueur de trame n'est pas compatible avec la valeur de longueur spécifiée dans le champ longueur/type. Si le champ longueur/type contient une valeur de type définie au 3.2.6 de l'ISO/CEI 8802-3:2000, la longueur de trame est considérée en adéquation avec ce champ et il convient de ce fait de ne pas considérer la DLPDU comme non valide;
- la longueur n'est pas un nombre entier d'octets;
- les bits de la DLPDU à l'arrivée, à l'exclusion du champ FCS, ne génèrent pas de valeur CRC identique à celle reçue;
- elle est incompatible avec une valeur de type F de DLPDU de bus de terrain de Type 21.

Le contenu de DLPDU non valides ne doit pas être transmis à l'utilisateur DL ou à la DLE. L'occurrence d'une DLPDU non valide peut être notifiée à la gestion du réseau.

NOTE Des DLPDU non valides peuvent être ignorées, rejetées ou utilisées de manière privée par un utilisateur DL autre que l'utilisateur DL du RTE. L'utilisation de telles DLPDU ne s'inscrit pas dans le domaine d'application de la présente norme.

6 Structure et procédure de DLPDU

6.1 Généralités

Le présent Article définit la structure, le contenu et le codage de chaque type et format de DLPDU, ainsi qu'un certain nombre d'éléments procéduraux. Les paragraphes décrivent la structure, le contenu, les paramètres et le codage des DLPDU, ainsi que la partie de la structure de DLPDU qui est spécifique au Type 21, comme illustré en Figure 4. Les aspects

relatifs à l'émission et à la réception par des utilisateurs DLS et par leurs DLE sont également décrits.

NOTE Dans le présent Article, toute référence au bit K de l'octet est une référence au bit dont le poids, dans un entier non signé d'un octet, est de 2^K ; ceci est parfois appelé numérotation binaire "petit-boutiste"..

6.2 Champ commun aux DLPDU

6.2.1 Généralités

La Figure 15 illustre le champ qui est commun aux DLPDU. Le champ version et longueur est commun à toutes les DLPDU et il n'est plus décrit pour chaque type de DLPDU.

Version and Length															
Version					Length										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Légende

Anglais	Français
Version and length	Version et Longueur
Version	Version
Length	Longueur

Figure 15 – Champ commun aux DLPDU

6.2.2 Version

Ce champ indique la version du protocole de Type 21 sur une séquence de 5 bits. Les 2 bits d'ordre supérieur indiquent la version majeure et les 3 bits d'ordre inférieur indiquent la version mineure (voir 5.3.3.1).

6.2.3 Longueur

Ce champ indique la longueur du champ de données en octets, y compris le champ FCS. La plage de valeurs admissibles est de 12 à 1 498.

6.3 Transfert de données DL

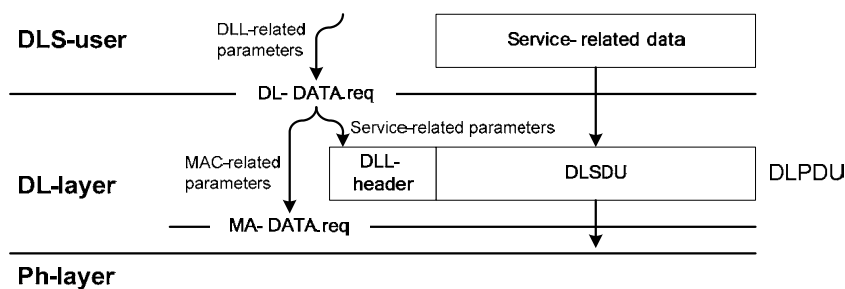
6.3.1 DLPDU de DT

6.3.1.1 Généralités

La DLPDU de DT est utilisée pour l'échange de données de Type 21 entre dispositifs.

6.3.1.2 Structure de DLPDU de DT

La Figure 16 illustre le processus de construction d'une DLPDU de DT et la Figure 17 illustre sa structure.



Légende

Anglais	Français
DLS-User	Utilisateur DLS
DLL-related parameters	Paramètres relatifs à la DLL
Service related data	Données relatives au service
DL-data-req	Demande de données DL
DL-layer	Couche DL
MAC-related parameters	Paramètres relatifs au MAC
Service-related parameters	Paramètres relatifs au service
DLL-header	En-tête DLL
MA-Data-req	Demande de données MA (activité du support)
Ph-layer	Couche physique

Figure 16 – Construction d'une DLPDU de DT

DST Addr	SRC Addr	FC	EXT	DSAP	SSAP	opt	DATA
----------	----------	----	-----	------	------	-----	------

Légende

Anglais	Français
DST addr	adr. Destinataire
SRC addr	adr. ORIGINE
DATA	DONNEES

Figure 17 – Structure d'une DLPDU de DT

6.3.1.3 Paramètres de DLPDU de DT

6.3.1.3.1 Généralités

Le Tableau 47 présente la liste des paramètres de DLPDU de DT.

Tableau 47 – Paramètres de DLPDU de DT

Paramètre	Type de Données	Valeur/Description
DST_addr	UNSIGNED16	Identifiant de l'entité DL de destination. [Facultatif] -Diffusion -Multidiffusion (définie par l'utilisateur) -Monodiffusion
Priorité	UNSIGNED8	0x00 à 0x03 0x04 à 0xFF: Réserve
DSAP	UNSIGNED16	DLSAP de destination de dispositif distant
SSAP	UNSIGNED16	DLSAP d'origine
Masque de groupe	—	Matrice binaire de dispositif de destination prévu. Uniquement en cas de diffusion ou de multidiffusion.
Longueur de masque de groupe	UNSIGNED8	Longueur de masque de groupe Uniquement en cas de diffusion ou de multidiffusion.
Données	—	DLSDU
Longueur de données	UNSIGNED16	Longueur de DLSDU

6.3.1.3.2 DST_addr

Ce paramètre indique l'identifiant de l'entité DL de destination de la ou des DLE auxquelles la trame est destinée. Il peut s'agir d'un identifiant d'entité DL individuel ou de multidiffusion (y compris la diffusion) et non de son adresse MAC. Il est explicitement attribué à l'adresse de destination pour une monodiffusion une valeur comprise entre 0x0000 et 0x00FF. Il est attribué à une adresse de multidiffusion définie par l'utilisateur une plage de valeurs allant de 0xFFDE à 0xFFFF. Il est attribué à une adresse de diffusion la valeur 0xFFFF. Les valeurs 0x0100 à 0xFFDC sont réservées. La valeur 0xFFDD indique une adresse non valide (voir 5.3.3.2).

6.3.1.3.3 Priorité

Ce champ indique la priorité de la trame sur une séquence à 2 bits. Ce champ contient la valeur du paramètre de priorité du message du service DL. La priorité la plus élevée est 0x03 et la plus basse est 0x00 (voir 5.3.3.4.3).

6.3.1.3.4 DSAP

Voir 5.3.3.6.

6.3.1.3.5 SSAP

Voir 5.3.3.7.

6.3.1.3.6 Masque de groupe

Voir 5.3.3.8.3.

6.3.1.3.7 Longueur de masque de groupe

Voir 5.3.3.8.2.

6.3.1.3.8 Données

Ce champ de données est la DLSDU.

6.3.1.3.9 Longueur de données

Ce champ indique la longueur du champ de données en octets. La plage de valeurs admissibles est de 0 à 1486.

6.3.1.4 Emission

Lorsque l'utilisateur DLS local lance une demande de service DL-DATA pour transférer une DLSDU à un utilisateur DLS homologue, la DLSDU est stockée dans la file d'attente RT et transmise par la DLPM au moyen d'une DLPDU de DT. L'option masque de groupe est disponible lorsque la DLPDU DL-DATA est diffusée ou multidiffusée.

Le Tableau 48 présente les primitives de service de liaison de données requises, ainsi que les paramètres pour l'émission d'une DLPDU de DT. Pour envoyer une DLPDU de DT à une DLE homologue, la DLPM lance une requête d'informations de trajet du dispositif homologue à la DLM en utilisant le service DLM-GET_PATH. La DLPDU de DT est alors transmise en utilisant une primitive de demande de service MA-DATA (données d'activité du support).

Tableau 48 – Primitives échangées entre un utilisateur DLS et une DLE pour envoyer une DLPDU de DT

Primitive	Origine	Paramètres associés
Demande DL-DATA	Utilisateur DLS	DST_addr DSAP SSAP Priorité Masque de groupe Longueur de masque de groupe Données Longueur de données
Demande DLM-GET_PATH	DLE (DLPM)	DST_addr
Confirmation DLM-GET_PATH	DLM	Etat Port R Adresse MAC
Demande MA-DATA	DLE (DLPM)	DLPDU Longueur de DLPDU

6.3.1.5 Réception

Lorsqu'une DLPDU de DT est reçue par l'intermédiaire de la couche MAC, la DLPM vérifie qu'elle est valide et qu'elle est destinée à l'utilisateur DLS local. Si la DLPDU de DT est diffusée ou multidiffusée, la DLPM examine le champ masque de groupe pour vérifier si la DLPDU de DT est destinée à l'utilisateur DLS local. Si la DLPDU est destinée à l'utilisateur DLS local, la DLPM extrait la DLSDU de la DLPDU de DT reçue et remet la DLSDU à l'utilisateur DLS approprié en utilisant une primitive d'indication DL-DATA. Si aucun utilisateur DLS n'est enregistré pour le DSAP dans la DLPDU de DT reçue, la DLPM rejette la DLPDU de DT qui vient d'être reçue.

Le Tableau 49 présente les primitives de service de liaison de données ainsi que les paramètres requis pour la réception d'une DLPDU de DT. Lorsqu'une DLPDU de DT est reçue

par l'intermédiaire d'une primitive d'indication MA-DATA, la DLPM extrait la DLSDU de la DLPDU de DT reçue et la remet à l'utilisateur DLS approprié.

Tableau 49 – Primitives échangées entre un utilisateur DLS et des DLE pour recevoir une DLPDU de DT

Primitive	Origine	Paramètres associés
Indication MA-DATA	MAC	DLPDU Longueur de DLPDU
Indication DL-DATA	DLPM	DLSDU Longueur de DLSDU

6.4 Transfert de DL-SPDATA (Données sporadiques de DL)

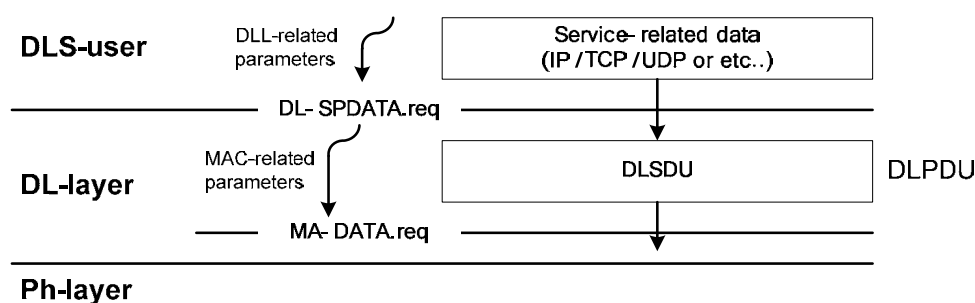
6.4.1 DLPDU de SPDT

6.4.1.1 Généralités

Les DLPDU de SPDT sont utilisées pour l'échange de données qui ne sont pas de Type 21, telles que des données IP.

6.4.1.2 Structure de DLPDU de SPDT

La Figure 18 illustre le processus de construction d'une DLPDU de SPDT et le Tableau 50 présente sa structure. L'en-tête de Type 21 n'est pas inclus dans la DLPDU de SPDT.



Légende

Anglais	Français
DLS-User	Utilisateur DLS
DLL-related parameters	Paramètres relatifs à la DLL
Service related data	Données relatives au service
(IP, TCP, UDP or etc.)	(IP, TCP, UDP ou etc.)
DL-SPDATA-req	demande de DL-SPDATA (données sporadiques de DL)
DL-layer	Couche DL
MAC-related parameters	Paramètres relatifs au MAC
MA-DATA-req	Demande de MA-DATA (activité du support)

Figure 18 – Structure de DLPDU de SPDT

6.4.1.3 Paramètres de DLPDU de SPDT

6.4.1.3.1 Généralités

Le Tableau 50 présente la liste des paramètres de la DLPDU de SPDT.

Tableau 50 – Paramètres de DLPDU de SPDT

Paramètre	Type de Données	Valeur/Description
Données	Chaîne d'octets	DLSDU
Longueur de données	UNSIGNED16	Longueur de DLSDU

6.4.1.3.2 Données

Ce champ indique la DLSDU qui doit être transmise dans une trame Ethernet conforme à l'ISO/CEI 8802-3:2000. Cette DLSDU peut être n'importe quelle unité de données de protocole, comme par exemple un paquet IP.

6.4.1.3.3 Longueur de données

Ce champ indique la longueur du champ de données en octets. La plage de valeurs admissibles est de 0 à 1 514.

6.4.1.4 Emission

Lorsque l'utilisateur DLS local lance une demande de service DL-SPDATA pour transférer une DLSDU à un utilisateur DLS homologue, la DLSDU est stockée dans la file d'attente NRT et transmise par la DLPM au moyen d'une DLPDU de SPDT. Dans ce cas, la DLSDU est transmise comme la DLPDU de SPDT sans en-tête de Type 21.

Le Tableau 51 présente les primitives de service de liaison de données requises, ainsi que les paramètres pour l'émission d'une DLPDU de SPDT. La DLPDU de SPDT est alors transmise par la DLPM en utilisant une primitive de service de demande MA-DATA.

Tableau 51 – Primitives échangées entre un utilisateur DLS et des DLE pour envoyer une DLPDU de SPDT

Primitive	Origine	Paramètres associés
Demande DL-SPDATA	Utilisateur DLS	Données: DLSDU Longueur de données: Longueur de DLSDU
Demande MA-DATA	DLPM	DLPDU Longueur de DLPDU
Confirmation MA-DATA	MAC	Etat

6.4.1.5 Réception

Lorsqu'une DLPDU de SPDT est reçue par l'intermédiaire de la couche MAC, la DLPM vérifie qu'elle est valide et qu'elle est destinée à l'utilisateur DLS local. Si c'est le cas, la DLPM extrait la DLSDU de la DLPDU de SPDT et la remet à l'utilisateur DLS approprié au moyen d'une primitive d'indication DL-DATA.

Le Tableau 52 présente les primitives de service de liaison de données requises, ainsi que les paramètres pour la réception d'une DLPDU de SPDT. Lorsqu'une DLPDU de SPDT est reçue par l'intermédiaire d'une primitive d'indication MA-DATA, la DLPM extrait la DLSDU de la DLPDU de SPDT reçue et la remet à l'utilisateur DLS approprié.

**Tableau 52 – Primitives échangées entre un utilisateur DLS
et des DLE pour recevoir une DLPDU de SPDT**

Primitive	Origine	Paramètres associés
Indication MA-DATA	DLPM	DLPDU Longueur de DLPDU
Indication DL-SPDATA	MAC	DLSDU Longueur de DLSDU

6.5 Messages de Contrôle du Réseau

6.5.1 Généralités

Les DLPDU de message de contrôle du réseau (NCM) sont utilisées pour les échanges de messages de contrôle du réseau entre DLM. Il est prévu cinq types de messages pour partager les informations du réseau.

6.5.2 DLPDU NCM_LA

6.5.2.1 Généralités

La DLPDU NCM_LA est utilisée pour transférer des informations de dispositif local aux dispositifs homologues lorsqu'une liaison est établie. Lorsque la DLM détecte le changement d'état de la liaison par l'intermédiaire de la primitive d'indication Ph-LINK_STATUS_CHANGE (CHANGEMENT D'ÉTAT DE LA LIAISON Ph), elle lance une requête d'information d'état de la liaison de la couche physique. La DLM génère ensuite une DLPDU NCM_LA pour envoyer les informations de liaison modifiées au dispositif homologue sur la liaison nouvellement établie.

6.5.2.2 Structure de DLPDU NCM_LA

Les messages NCM reposent sur la structure de base de la DLPDU de Type 21 illustrée en Figure 19. La Figure 19 présente la structure de la DLPDU NCM_LA après l'en-tête Ethernet.

DST Addr (C_NCM_ADDR)	SRC Addr	FC	EXT	DSAP	SSAP	DATA (Local Device Information)
--------------------------	----------	----	-----	------	------	------------------------------------

Légende

Anglais	Français
DST addr (C_NCM_Addr)	adr. de DESTINATION (C_NCM_Addr)
SRC Addr	adr. ORIGINE
DATA (local device information)	DONNEES (Informations de dispositif local)

Figure 19 – Structure de DLPDU NCM_LA

6.5.2.3 Paramètres de DLPDU NCM_LA

Le Tableau 53 présente la liste des paramètres de la DLPDU NCM_LA.

Tableau 53 – Paramètres de DLPDU NCM_LA

Paramètre	Type de Données	Description
DST_addr	UNSIGNED16	C_NCM_ADDR (voir 5.3.3.2)
SRC_addr	UNSIGNED16	DL_ADDR d'identifiant d'entité DL locale (voir 5.3.3.3)
NCMT	UNSIGNED8	NCM_FAMILY_REQ (voir 5.3.3.4.5.2)
DLMDU	—	Unité de données de gestion DL Informations de dispositif local (voir 4.6.5)
Longueur	UNSIGNED16	Longueur de DLMDU
Port R	UNSIGNED8	Port R de liaison nouvellement activée

6.5.2.4 Emission

Lorsque la DLM détecte le changement d'état de la liaison par l'intermédiaire de la primitive d'indication Ph-LINK_STATUS_CHANGE, elle lance une requête d'information d'état de la liaison de la couche physique. Si la liaison est nouvellement établie, la DLM génère une DLPDU NCM_LA pour envoyer les informations de liaison modifiées au dispositif homologue sur la liaison nouvellement établie. La DLPDU NCM_LA est stockée dans la file d'attente RT et transmise par la DLPM au moyen de la primitive de requête MA-DATA.

6.5.2.5 Réception

L'adresse C_NCM_ADDR est utilisée comme identifiant d'entité DL de destination dans la DLPDU NCM_LA. Lorsqu'une DLPDU NCM_LA est reçue par l'intermédiaire de la couche MAC, la DLM incrémente le champ Nombre de sauts dans la DLPDU et achemine la DLPDU par l'autre Port R. La DLPDU NCM_LA est ignorée par la LNM ou par le dispositif qui a généré la DLPDU. La DLPDU NCM_LA est reçue et traitée par la DLM de la manière suivante:

- la DLM s'assure que la DLPDU est générée par le dispositif lui-même. Si c'est le cas et si la DLPDU est reçue par l'intermédiaire de l'autre Port R, ceci signifie que le réseau est configuré en topologie annulaire. Dans ce cas, on passe à l'étape f) et on termine la procédure par l'étape g). Sinon, on passe à l'étape b);
- mettre le type de dispositif connecté du Port R dans PORT_INFO (voir 4.6.5.9) sur "type de dispositif homogène";
- extraire les informations de trajet du dispositif (DPI) de la DLMDU et vérifier une éventuelle collision d'identifiant d'entité DL en utilisant les DPI. Si l'adresse d'origine n'est pas enregistrée dans les DPI, générer un événement EVENT_IN_DEVICE. S'il est détecté une collision d'identifiant d'entité DL locale, établir DEV_FLAG (voir 4.6.5.3) sur "collision d'identifiant d'entité DL locale" et générer un événement EVENT_THIS_ADDR_COLLISION. S'il est détecté une collision d'identifiant d'entité DL de réseau, établir NET_FLAG (voir 4.6.6.11) sur "état de collision d'identifiant d'entité DL de réseau" et générer un événement EVENT_NET_ADDR_COLLISION. La notification de tout événement généré est envoyée à l'utilisateur DLMS local en utilisant la primitive de service d'indication DLM-EVENT;
- incrémenter HOP_CNT (voir 4.6.5.13) dans la DLMDU reçue;
- acheminer la DLPDU NCM_LA modifiée par l'autre Port R;
- vérifier la condition d'événement de déclenchement d'état DLM en utilisant PORT_INFO et effectuer la transition d'état de la DLM (voir 7.3.3);
- générer une DLPDU NCM_AT et la transmettre par le Port R opposé.

6.5.3 DLPDU NCM_AT

6.5.3.1 Généralités

Une DLPDU NCM_AT est utilisée pour transférer les informations de dispositif local vers les autres dispositifs du réseau lorsqu'un dispositif reçoit une DLPDU NCM_LA. Cette DLPDU est transmise par le Port R qui est utilisé pour recevoir la DLPDU NCM_LA.

6.5.3.2 Structure de DLPDU NCM_AT

Voir 6.5.2.2.

6.5.3.3 Paramètres de DLPDU NCM_AT

Le Tableau 54 présente la liste des paramètres de la DLPDU NCM_AT.

Tableau 54 – Paramètres de DLPDU NCM_AT

Paramètre	Type de Données	Description
DST_addr	UNSIGNED16	C_NCM_ADDR (voir 5.3.3.2.3)
SRC_addr	UNSIGNED16	DL_ADDR d'identifiant d'entité DL locale (voir 4.6.5.2)
NCMT	UNSIGNED8	NCM_ADV_THIS (voir 4.4.3.2.4 de la CEI 61158-3-21:2010)
DLMDU	—	Informations de dispositif local (voir 4.6.5)
Longueur	UNSIGNED16	Longueur de DLMDU
Port R	UNSIGNED8	Port R utilisé pour la réception de la DLPDU NCM_LA

6.5.3.4 Emission

Lorsque la DLM reçoit une DLPDU NCM_LA, elle génère une DLPDU NCM_AT pour la DLMDU, comprenant les informations de dispositif local. La DLMDU est enregistrée dans la file d'attente RT et transmise par la DLPM par l'intermédiaire du Port R d'où la DLPDU NCM_LA a été reçue. Dans ce cas, C_NCM_ADDR est utilisé comme identifiant d'entité DL de destination.

6.5.3.5 Réception

L'adresse C_NCM_ADDR est utilisée comme adresse de destination dans une DLPDU NCM_AT. Lorsqu'une DLPDU NCM_AT est reçue par l'intermédiaire de la couche MAC, la DLM incrémente le champ Nombre de sauts dans la DLPDU et achemine la DLPDU par l'autre Port R. La DLPDU NCM_AT est ignorée par la LNM ou par le dispositif qui a généré la DLPDU. La DLPDU NCM_AT est reçue et traitée par la DLM de la manière suivante:

- la DLM s'assure que la DLPDU a été générée par le dispositif lui-même. Si la DLPDU a été générée localement et reçue par l'intermédiaire de l'autre Port R, ceci signifie que le réseau est configuré en topologie annulaire. Dans ce cas, on passe à l'étape f). Sinon, on passe à l'étape b);
- mettre le type de dispositif connecté du Port R dans PORT_INFO (voir 4.6.5.9) sur "type de dispositif homogène";
- extraire les informations de trajet du dispositif (DPI) (voir 4.6.7) de la DLMDU et vérifier une éventuelle collision d'identifiant d'entité DL en utilisant les DPI. Si l'adresse d'origine n'est pas enregistrée dans les DPI, générer un événement EVENT_IN_DEVICE. S'il est détecté une collision d'identifiant d'entité DL locale, établir DEV_FLAG (voir 4.6.5.3) sur "collision d'identifiant d'entité DL locale" et générer un événement EVENT_THIS_ADDR_COLLISION. S'il est détecté une collision d'identifiant d'entité DL de réseau, établir NET_FLAG (voir 4.6.6.11) sur "état de collision d'identifiant d'entité DL de réseau" et générer un événement EVENT_NET_ADDR_COLLISION. La notification de

tout événement généré est envoyée à l'utilisateur DLMS local au moyen de la primitive de service d'indication DLM-EVENT;

- d) incrémenter HOP_CNT (voir 4.6.5.13) dans la DLMDU reçue;
- e) acheminer la DLPDU NCM_LA modifiée par l'autre Port R;
- f) vérifier la condition d'événement de déclenchement d'état DLM en utilisant PORT_INFO et effectuer la transition d'état de la DLM (voir 7.3.3).

6.5.4 DLPDU NCM_LS

6.5.4.1 Généralités

La DLPDU NCM_LS est utilisée pour indiquer que le réseau est automatiquement configuré en topologie linéaire lorsqu'un réseau linéaire est nouvellement établi ou qu'un réseau linéaire existant a été divisé en deux lignes, ou encore qu'un réseau annulaire a été reconfiguré en réseau linéaire. Cette DLPDU est générée par le dispositif choisi comme LNM sur le réseau.

6.5.4.2 Structure de DLPDU NCM_LS

Voir 6.5.2.2.

6.5.4.3 Paramètres de DLPDU NCM_LS

Le Tableau 55 présente la liste des paramètres de la DLPDU NCM_LS.

Tableau 55 – Paramètres de DLPDU NCM_LS

Paramètre	Type de Données	Description
DST_addr	UNSIGNED16	Adresse de diffusion (voir 5.3.3.2.2)
SRC_addr	UNSIGNED16	DL_ADDR d'identifiant d'entité DL locale (voir 4.6.5.2)
NCMT	UNSIGNED8	NCM_LINE_START (voir 5.3.3.4.5.4)
DLMDU	—	Informations de dispositif local (voir 4.6.5)
Longueur	UNSIGNED16	Longueur de DLMDU
Port R	UNSIGNED8	Port R de liaison activée

6.5.4.4 Emission

Lorsque l'état de la DLM passe à LNM, la DLM génère une DLPDU NCM_LS pour informer chaque dispositif du réseau de la modification de la topologie du réseau. La DLPDU NCM_LS est stockée dans la file d'attente RT et diffusée par la DLPM au moyen de la primitive de demande MA-DATA.

6.5.4.5 Réception

L'identifiant d'entité DL diffusé est utilisé comme adresse de destination dans une DLPDU NCM_LS. Par conséquent, la DLPDU NCM_LS est directement acheminée par la couche MAC. La DLPDU NCM_LS est traitée par la DLM de la manière suivante:

- a) extraire les DPI (voir 4.6.7) de la DLMDU et les utiliser pour vérifier une éventuelle collision d'identifiant d'entité DL. Si l'adresse d'origine n'est pas enregistrée dans les DPI, générer un événement EVENT_IN_DEVICE. S'il est détecté une collision d'identifiant d'entité DL locale, établir DEV_FLAG (voir 4.6.5.3) sur "collision d'identifiant d'entité DL locale" et générer un événement EVENT_THIS_ADDR_COLLISION. S'il est détecté une collision d'identifiant d'entité DL de réseau, établir NET_FLAG (voir 4.6.6.11) sur "état de collision d'identifiant d'entité DL de réseau" et générer un événement

EVENT_NET_ADDR_COLLISION. La notification de tout événement généré est envoyée à l'utilisateur DLMS local au moyen de la primitive de service d'indication DLM-EVENT;

- b) vérifier la condition d'événement de déclenchement d'état DLM en utilisant PORT_INFO et effectuer la transition d'état de la DLM (voir 7.3.3);
- c) si l'état DLM n'est pas LNM ou RNM, activer les fonctions d'acheminement de trame et établir sur VRAI la "fonction d'acheminement des trames du Port R1 au Port R2" et la "fonction d'acheminement des trames du Port R2 au Port R1" dans PORT_INFO (voir 4.6.5.9).

6.5.5 DLPDU NCM_RS

6.5.5.1 Généralités

La DLPDU NCM_RS est utilisée pour indiquer que le réseau est automatiquement configuré en réseau annulaire. Cette DLPDU est générée par le dispositif choisi comme RNMP sur le réseau.

6.5.5.2 Structure de DLPDU NCM_RS

Voir 6.5.2.2.

6.5.5.3 Paramètres de DLPDU NCM_RS

Le Tableau 56 présente la liste des paramètres de la DLPDU NCM_RS.

Tableau 56 – Paramètres de DLPDU NCM_RS

Paramètre	Type de Données	Description
DST_addr	UNSIGNED16	Adresse de diffusion
SRC_addr	UNSIGNED16	DL_ADDR d'identifiant d'entité DL locale (voir 4.6.5.2)
NCMT	UNSIGNED8	NCM_RING_START (voir 4.4.3.2.4 de la CEI 61158-3-21:2010)
DLMDU	—	Informations de dispositif local (voir 4.6.5)
Longueur	UNSIGNED16	Longueur de DLMDU
Port R	UNSIGNED8	Port R1, Port R2

6.5.5.4 Emission

Lorsque l'état de la DLM passe à RNMP, la DLM génère une DLPDU NCM_RS pour informer chaque dispositif du réseau de la modification de la topologie du réseau. La DLPDU NCM_RS est stockée dans la file d'attente RT et diffusée par la DLPM au moyen de la primitive de demande MA-DATA.

6.5.5.5 Réception

L'identifiant d'entité DL diffusé est utilisé comme adresse de destination dans une DLPDU NCM_RS. Par conséquent, la DLPDU NCM_RS est directement acheminée par la couche MAC. La DLPDU NCM_RS est traitée par la DLM de la manière suivante:

- a) extraire les DPI (voir 4.6.7) de la DLMDU et les utiliser pour vérifier une éventuelle collision d'identifiant d'entité DL. Si l'adresse d'origine n'est pas enregistrée dans les DPI, générer un événement EVENT_IN_DEVICE. S'il est détecté une collision d'identifiant d'entité DL locale, établir DEV_FLAG (voir 4.6.5.3) sur "collision d'identifiant d'entité DL locale" et générer un événement EVENT_THIS_ADDR_COLLISION. S'il est détecté une collision d'identifiant d'entité DL de réseau, établir NET_FLAG (voir 4.6.6.11) sur "état de collision d'identifiant d'entité DL de réseau" et générer un événement EVENT_NET_ADDR_COLLISION. La notification de tout événement généré est envoyée à l'utilisateur DLMS local au moyen de la primitive de service d'indication DLM-EVENT;

- b) vérifier la condition d'événement de déclenchement d'état DLM en utilisant PORT_INFO et effectuer la transition d'état de la DLM (voir 7.3.3);
- c) si l'état DLM n'est pas RNMP ou RNMS, activer les fonctions d'acheminement des trames et établir sur "VRAI" la "fonction d'acheminement de trame du Port R1 au Port R2" et la "fonction d'acheminement des trames du Port R2 au Port R1" dans PORT_INFO (voir 4.6.5.9).
- d) lorsque le dispositif est désigné comme étant le RNMS par le RNMP, désactiver la fonction d'acheminement des trames dans le sens RNMP;
- e) passer à l'état RNMS si le contrôle d'acheminement des trames s'est achevé avec succès. Sinon, rester à l'état GD;
- f) lorsque la DLM passe à l'état RNMS, envoyer NCM_ACK_RNMS au RNMP.

6.5.6 DLPDU NCM_AR

6.5.6.1 Généralités

La DLPDU NCM_AR est utilisée pour indiquer que le RNMS désigné a été attribué avec succès et qu'il est opérationnel. Cette DLPDU est générée par le dispositif choisi comme RNMS et transférée au RNMP.

6.5.7 Structure de DLPDU NCM_AR

6.5.7.1 Généralités

Voir 6.5.2.2.

6.5.7.2 Paramètres de DLPDU NCM_AR

Le Tableau 57 présente la liste des paramètres de la DLPDU NCM_AR.

Tableau 57 – Paramètres de DLPDU NCM_AR

Paramètre	Type de Données	Description
DST_addr	UNSIGNED16	Identifiant d'entité DL de RNMP
SRC_addr	UNSIGNED16	DL_ADDR d'identifiant d'entité DL locale (voir 4.6.5.2)
NCMT	UNSIGNED8	NCM_ACK_RNMS (voir 4.4.3.2.4 de la CEI 61158-3-21:2010)
DLMDU	—	Informations de dispositif local (voir 4.6.5)
Longueur	UNSIGNED16	Longueur de DLMDU
Port R	UNSIGNED8	Port R de destination dans la table des trajets

6.5.7.3 Emission

Lorsque l'état de la DLM passe de GD à RNMS, la DLM génère une DLPDU NCM_AR pour indiquer que le RNMS désigné a été attribué avec succès et effectue une monodiffusion de cette DLPDU destinée au RNMP.

6.5.7.4 Réception

La DLPDU NCM_AR fait l'objet d'une monodiffusion du RNMS au RNMP. La DLPDU NCM_RS est traitée par la DLM dans le dispositif RNMP de la manière suivante:

- a) extraire les DPI (voir 4.6.7) de la DLMDU et les utiliser pour vérifier une éventuelle collision d'identifiant d'entité DL. Si l'adresse d'origine n'est pas enregistrée dans les DPI, générer un événement EVENT_IN_DEVICE. S'il est détecté une collision d'identifiant d'entité DL locale, établir DEV_FLAG (voir 4.6.5.3) sur "collision d'identifiant d'entité DL locale" et générer un événement EVENT_THIS_ADDR_COLLISION. S'il est détecté une

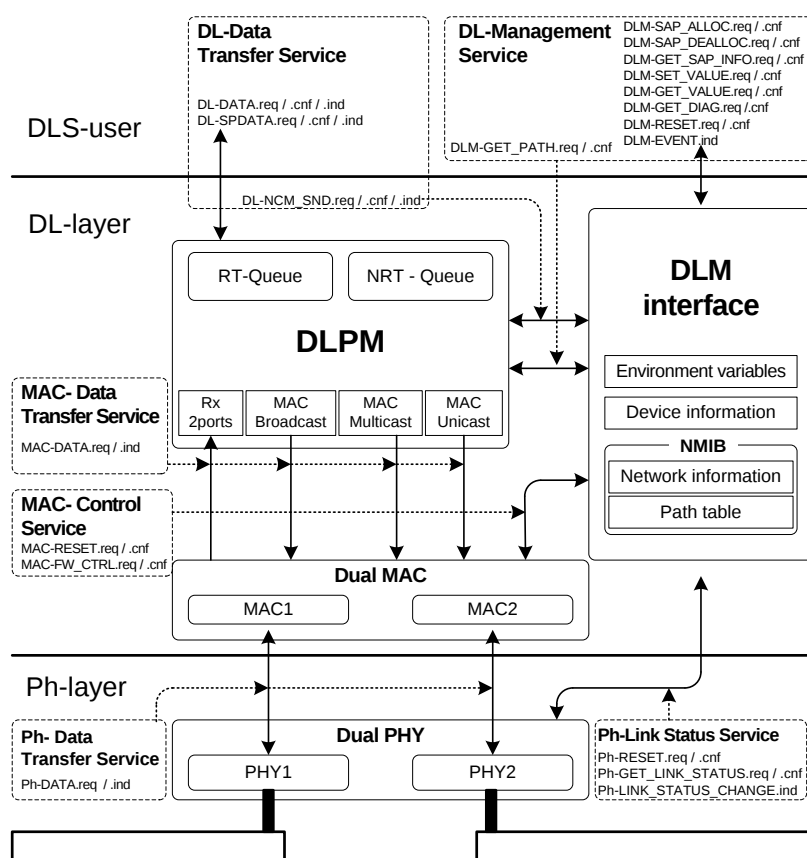
collision d'identifiant d'entité DL de réseau, établir NET_FLAG (voir 4.6.6.11) sur "état de collision d'identifiant d'entité DL de réseau" et générer un événement EVENT_NET_ADDR_COLLISION. La notification de tout événement généré est envoyée à l'utilisateur DLMS local au moyen de la primitive de service d'indication DLM-EVENT;

- b) Désactiver la fonction d'acheminement des trames dans le sens RNMS.

7 Eléments de procédure de DLE

7.1 Structure globale

La DLL est composée des éléments de contrôle de la DLPM, du MAC double (DMAC), de l'interface physique double (DPHY) et de l'interface de gestion DLL. La DLPM est l'élément de commande principal. Elle assure les fonctions de MAC déterministe en coordonnant le DMAC et les messages NCM pour une prise en charge fiable et efficace des services de niveau supérieur de transfert de données temps réel et non temps réel, en mode sans connexion. L'interface de gestion DLL assure les fonctions de gestion de la DLL. La structure globale de la DLL est décrite en Figure 20.



Légende

Anglais	Français
DLS-user	Utilisateur DLS
DL-Data transfer service	Service de transfert de données DL
DL-management service	Service de gestion DL
DL-layer	Couche DL
RT-queue	File d'attente RT
NRT-queue	File d'attente NRT
DLM interface	Interface DLM

Anglais	Français
MAC-data transfer service	Service de transfert de données MAC
Rx 2 ports	2 ports Rx
MAC broadcast	Diffusion MAC
MAC multicast	Multidiffusion MAC
MAC unicast	Monodiffusion MAC
Environment variables	Variables d'environnement
Device information	Informations de dispositif
Network information	Informations de réseau
Path table	Table des trajets
MAC control service	Service de contrôle MAC
Dual MAC	MAC double
Ph-Layer	Couche physique
Ph-data transfer service	Service de transfert de données Ph
Dual PHY	PHY double
Ph-link status service	Service d'état de liaison Ph

Figure 20 – Structure et éléments de DLL

7.2 Machine protocolaire DL (DLPM)

7.2.1 Vue d'ensemble

La DLPM maintient deux files d'attente d'émetteur et une file d'attente de récepteur. Lorsqu'un utilisateur DLS local ou un utilisateur DLMS génère un message, qui est stocké dans la file d'attente RT ou dans la file d'attente NRT sous la forme d'une DLPDU. La DLPM traite les messages dans la file d'attente RT en fonction de la DLPMS. La DLPM transmet le message en utilisant la primitive de demande MA-DATA. Lorsqu'un message est reçu par la couche MAC, il est stocké dans la file d'attente de réception et traité en FIFO (premier entré, premier sorti) par la file d'attente.

7.2.2 Définition des primitives

7.2.2.1 Primitives échangées entre DLPM et utilisateur DLS

Le Tableau 58 présente les primitives de service de liaison de données échangées entre la DLPM et l'utilisateur DLS.

Tableau 58 – Primitives échangées entre DLPM et utilisateur DLS

Primitive	Origine	Paramètres associés	Description
DL-DATA.req	Utilisateur DLS	DST_addr DSAP SSAP Priorité DLSDU Longueur de DLSDU	Demande d'émission vers un dispositif distant de Type 21
DL-DATA.cnf	DLPM	Etat	Confirmation destinée à l'utilisateur DLS appelant
DL-DATA.ind	DLPM	DST_addr SRC_addr SSAP DLSDU Longueur de DLSDU	Indication de réception provenant d'un dispositif distant de Type 21
DL-SPDATA.req	Utilisateur DLS	DLSDU Longueur de DLSDU	Demande d'émission de données sporadiques vers un dispositif distant de Type 21
DL-SPDATA.cnf	DLPM	Etat	Confirmation destinée à l'utilisateur DLS appelant
DL-SPDATA.ind	DLPM	DLSDU Longueur de DLSDU	Indication de réception de données sporadiques provenant d'un dispositif distant de Type 21

Le Tableau 59 présente les paramètres échangés entre la DLPM et l'utilisateur DLS.

Tableau 59 – Paramètres échangés entre DLPM et utilisateur DLS

Paramètre	Description
DST_addr	Identifiant de l'entité DL de destination. C_BROADCAST_ADDR: adresse de diffusion 0xFFFF Adresse de multidiffusion définie par l'utilisateur: 0xFFFD à 0xFFDE 0-MAX_DEVICE_ADDR: adresse monodestinataire
DSAP	Point d'accès de service de destination
SSAP	Point d'accès de service d'origine
Priorité	Priorité du message. Les valeurs admissibles sont C_PRI_0 à C_PRI_3 C_PRI_3 est la priorité la plus élevée.
DLSDU	DLSDU
Longueur de DLSDU	Longueur de DLSDU
Etat	Ce paramètre permet à l'utilisateur DLMS de déterminer si le DLMS demandé a été fourni avec succès. En cas d'échec, la raison est spécifiée. Ce paramètre prend l'une des valeurs suivantes: "OK - succès - la variable a pu être mise à jour"; "Défaillance - la variable n'existe pas ou n'a pas pu prendre la nouvelle valeur"; "Echec - paramètres invalides dans la demande".

NOTE C_NCM_ADDR n'est pas utilisé pour des services DL-DATA et DL-SPDATA.

7.2.2.2 Primitives échangées entre DLPM et DLM

Le Tableau 60 et le Tableau 61 présentent les primitives de service de liaison de données et paramètres échangés entre la DLPM et la DLM.

Tableau 60 – Primitives échangées entre DLPM et DLM

Primitive	Origine	Paramètres associés	Description
DLM-GET_PATH.req	DLPM	DST_addr	Demande d'informations de trajet vers la DLM
DLM-GET_PATH.cnf	DLM	Etat Port R Adresse MAC	Confirmation destinée à la DLPM appelante avec Port R et adresse MAC
DLM-NCM_SND.req	DLM	DST_addr NCMT DLMDU Longueur Port R	Demande d'émission de Message de contrôle du réseau NCM_FAMILY_REQ NCM_FAMILY_RES NCM_MEDIA_LINKED NCM_ADV_THIS NCM_LINE_START NCM_RING_START NCM_ACK_RNMS NCM_RETRY_RNMS
DLM-NCM_SND.cnf	DLPM	Etat	Confirmation destinée à l'utilisateur du service appelant
DLM-NCM_SND.ind	DLPM	DST_addr SRC_addr NCMT DLMDU longueur Port R	Indication de réception de Message de contrôle du réseau provenant d'un dispositif distant de Type 21

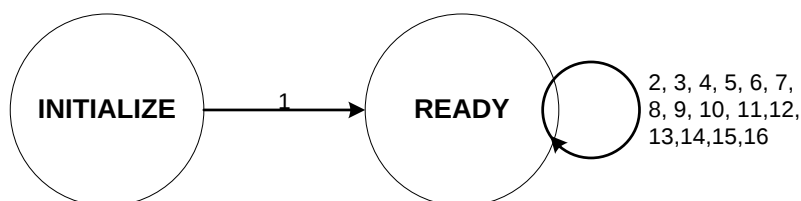
Tableau 61 – Paramètres utilisés avec des primitives échangées entre DLPM et DLM

Paramètre	Description
DST_addr	Identifiant de l'entité DL de destination. C_BROADCAST_ADDR: adresse de diffusion, 0xFFFF C_NCM_ADDR: Identifiant d'entité DL fixe pour Message de contrôle du réseau, 0xFFFE 0-MAX_DEVICE_ADDR: adresse monodestinataire
SRC_addr	Identifiant d'entité DL d'origine. 0-MAX_DEVICE_ADDR: adresse monodestinataire
Port R	Port R utilisé pour émettre ou recevoir une trame.
Adresse MAC	Adresse MAC de l'ISO/CEI 8802:3:2000 qui correspond à l'identifiant d'entité DL de destination.
NCMT	Type de Message de contrôle du réseau. NCM_FAMILY_REQ (0x01), NCM_FAMILY_RES (0x02), NCM_MEDIA_LINKED (0x03), NCM_ADV_THIS (0x04), NCM_LINE_START (0x05), NCM_RING_START (0x06), NCM_ACK_RNMS (0x07), NCM_RETRY_RNMS (0x08)
DLMDU	Unité de données de gestion DL. Le champ de données du Message de contrôle du réseau DLM.
Longueur	Longueur de DLMDU
Etat	Ce paramètre permet à l'utilisateur DLMS de déterminer si le DLMS demandé a été fourni avec succès. En cas d'échec, la raison est spécifiée. Ce paramètre doit prendre l'une des valeurs suivantes: "OK - succès - la variable a pu être mise à jour"; "Défaillance - la variable n'existe pas ou n'a pas pu prendre la nouvelle valeur"; "Echec - paramètres invalides dans la demande".

NOTE Les adresses de multidiffusion définies par l'utilisateur ne sont pas utilisées pour l'identifiant d'entité DL de destination.

7.2.3 Table d'états DLPM

La Figure 21 décrit le diagramme des transitions d'états de la DLPM. La table d'états de la DLPM est donnée dans le Tableau 62.



Légende

Anglais	Français
Initialize	Initialiser
Ready	Prêt

Figure 21 – Diagramme des transitions d'états de la DLPM

Tableau 62 – Table d'états DLPM

"#	Courant	Événement / Condition =>actions	Etat suivant
1	INITIALIZE	POWER-ON or RESET / =>	READY
2	READY	DL-DATA.req {DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length } / CHECK_PARA(DLSDU length) = "True" && DLM-GET_PATH.req {DST_addr} Status:= DLM-GET_PATH.cfm { D_MAC_addr, R-port } Status = "Success" && CHECK_RTQUEUE() <> "Full" => DLPDU:= BUILD_DLPDU(D_MAC_addr, DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length) ENQUEUE_RT(Priority, DLPDU, DLPDU length, R-port) DL-DATA.cfm{Status:= "Success"}	READY
3	READY	DL-DATA.req {DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length } / CHECK_PARA(DLSDU length) <> " True" => DL-DATA.cfm{Status:= "Failure – Invalid parameter"}	READY
4	READY	DL-DATA.req {DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length } / CHECK_PARA(DLSDU length) = " True" && DLM-GET_PATH.ind { D_MAC_addr, R-port} ="Failure" => DL-DATA.cfm{Status:= "Failure – not available destination"}	READY
5	READY	DL-DATA.req { DST_addr, DSAP, SSAP, Priority, DLSDU, DLSDU length } / CHECK_PARA(DLSDU length) = " True" && DLM-GET_PATH.ind { D_MAC_addr, R-port } =" Success" && CHECK_RTQUEUE() = "Full" => DL-DATA.cfm{Status:= "Failure – The RT-queue is full"}	READY

"#	Courant	Événement / Condition =>actions	Etat suivant
6	READY	DL-SPDATA.req {DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = "True" && CHECK_NRTQUEUE() <> "Full" => DLPDU:= DLSDU R-port:= R-port1 && R-port2 ENQUEUE_NRT(DLPDU, DLPDU length, R-port) DL-SPDATA.cfm {Status:= "Success"}	READY
7	READY	DL-SPDATA.req {DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = "False" => DL- SPDATA.cfm{Status:= "Failure – Invalid requested parameter"}	READY
8	READY	DL-SPDATA.req {DLSDU, DLSDU length} / CHECK_PARA(DLSDU length) = "True" && CHECK_NRTQUEUE() = "Full" => DL- SPDATA.cfm{Status:= "Failure – The NRT-queue is full"}	READY
9	READY	DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} / CHECK_PARA(Length) = "True" && CHECK_RTQUEUE() <> "Full" => Priority:= C_HIGHEST_PRIORITY DLPDU:= BUILD_DLPDU_NCM(DST_addr , Priority, NCMT , DLMDU, Length) QUEUE_RT(Priority, DLPDU, DLPDU length, R-port) DL-NCM_SND.cfm{Status:= "Success"}	READY
10	READY	DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} / CHECK_PARA(Length) = "False" => DL- NCM_SND.cfm{Status:= "Failure – Invalid parameter"}	READY
11	READY	DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} / CHECK_PARA(Length) = "True" && CHECK_RTQUEUE() = "Full" => DL- NCM_SND.cfm{Status:= "Failure – The RT-queue is full"}	READY

"#	Courant	Événement / Condition =>actions	Etat suivant
12	READY	MA-DATA.ind{DLPDU, DLPDU length, R-port} / CHECK_FRAME_ETH_TYPE(DLPDU) = "Type 21" && CHECK_FRAME_FOR_THIS(DLPDU) = "True" && CHECK_FC(DLPDU) <> "Network control message" => DST_addr:= GET_D_ADDR(DLPDU) SRC_addr:= GET_S_ADDR(DLPDU) DSAP:= GET_DSAP(DLPDU) SSAP:= GET_SSAP(DLPDU) DLSU:= GET_DLSU(DLPDU) DLSU length:= GET_DLSU(DLPDU length) DLS-user:= GET_DLS_USER(DSAP) dst_dls_user := SET_DLS_USER(DLS-user) dst_dls_user.DL-DATA.ind{ DST_addr, SRC_addr, SSAP, DLSU, DLSU length}	READY
13	READY	MA-DATA.ind{DLPDU, DLPDU length, R-port} / CHECK_FRAME_ETH_TYPE(DLPDU) <> "Type 21" => DLSU:= DLPDU DLSU length:= DLPDU length DL-SPDATA.ind{DLSU, DLSU length}	READY
14	READY	MA-DATA.ind{DLPDU, DLPDU length, R-port} / CHECK_FRAME_ETH_TYPE(DLPDU) = "Type 21" && CHECK_FRAME_FOR_THIS(DLPDU) = "True" && CHECK_FC(DLPDU) = "Network Control Message" => DST_addr:= GET_D_ADDR(DLPDU) SRC_addr:= GET_S_ADDR(DLPDU) DLMDU:= GET_DLSU(DLPDU) length:= GET_DLSU(DLPDU length) cmd:= GET_NCM_CMD(DLPDU) DL-NCM_SND.ind{DST_addr, SRC_addr, NCMT, DLMDU, DLMDU length, R-port}	READY
15	READY	New DLPDU from DLS-user / CHECK_RTQUEUE() <> "Empty" => DLPDU:= DQUEUE_RT(policy) MA-DATA.req {DLPDU, DLPDU length, R-port}	READY

"#	Courant	Événement / Condition =>actions	Etat suivant
16	READY	New DLPDU from DLS-user / CHECK_RTQUEUE() = "Empty" && / CHECK_NRTQUEUE() <> "Empty" => DLPDU:= DQUEUE_NRT() MA-DATA.req {DLPDU, DLPDU length, R-port}	READY

7.2.4 Fonctions DLPM

Toutes les fonctions de la DLPM sont résumées dans le Tableau 63.

Tableau 63 – Table des fonctions DLPM

Nom de la fonction	Entrée	Sortie	Fonctionnement
CHECK_PARA	Longueur de DLSDU	Vrai/Faux	Renvoyer "FAUX" si la taille de la DLSDU dépasse C_MAX_USER_DLMDU_SIZE. Sinon, renvoyer "VRAI".
CHECK_RTQUEUE	(aucune)	état	Vérifier la condition de la file d'attente RT. L'état renvoyé est "Pleine," "Vide" ou "File d'attente".
BUILD_DLPDU	D_MAC_addr DST_addr DSAP SSAP Priorité DLSDU Longueur de DLSDU	DLPDU Longueur de DLPDU	Construire la DLPDU et renvoyer la DLPDU et sa longueur.
ENQUEUE_RT	Priorité DLPDU Longueur de DLPDU Port R	(aucune)	Met les données d'entrée en queue de la file d'attente RT.
CHECK_NRTQUEUE	(aucune)	état	Vérifier que la condition de la file d'attente NRT pour les DATA est "pleinement en file d'attente". L'état renvoyé est "Pleine," "Vide" ou "File d'attente".
ENQUEUE_NRT	DLPDU Longueur de DLPDU Port R	(aucune)	Met les données d'entrée en file d'attente NRT sur une base FIFO.
BUILD_DLPDU_NCM	DST_addr Priorité NCMT DLMDU Longueur de DLMDU	DLPDU Longueur de DLPDU	Construire la DLPDU de NCM et renvoyer la DLPDU et sa longueur.
DQUEUE_RT	Stratégie	DLPDU Longueur de DLPDU Port R	Obtenir une DLPDU de la file d'attente RT comme spécifié par la DLPMSF.
DQUEUE_NRT	(aucune)	DLPDU Longueur de DLPDU Port R	Obtenir une DLPDU de la file d'attente NRT selon la méthode FIFO.
CHECK_FRAME_ETH_TYPE	DLPDU	état	Vérifier que le type Ethernet est le Type 21 (0x88FE). Renvoyer "Type 21" si le type est correct. Sinon, renvoyer "Autre".

Nom de la fonction	Entrée	Sortie	Fonctionnement
CHECK_FRAME_FOR_THIS	DLPDU	Vrai/Faux	Vérifier si la trame a été envoyée en monodiffusion au dispositif local. Renvoyer "VRAI" si l'identifiant d'entité DL de destination est égal à l'identifiant d'entité DL locale. Sinon, renvoyer "FAUX".
CHECK_FC	DLPDU	état	Vérifier si la trame reçue est un Message de contrôle du réseau. Renvoyer "Message de contrôle du réseau" si le champ FC indique NCM. Sinon, renvoyer "Autre".
GET_D_ADDR	DLPDU	DST_addr	Renvoyer l'identifiant d'entité DL de destination de la DLPDU.
GET_S_ADDR	DLPDU	SRC_addr	Renvoyer l'identifiant d'entité DL d'origine de la DLPDU.
GET_DSAP	DLPDU	DSAP	Renvoyer le SAP de destination de la DLPDU.
GET_SSAP	DLPDU	SSAP	Renvoyer le SAP d'origine de la DLPDU.
GET_DLSDU	DLPDU	DLSDU	Renvoyer la DLSDU de la DLPDU.
GET_DLSDU LENGTH	Longueur de DLPDU	Longueur de DLSDU	Renvoyer la longueur de la DLSDU.
GET_DLS_USER	DSAP	ID d'utilisateur DLS	Renvoyer l'ID d'utilisateur DLS qui possède le DSAP.
SET_DLS_USER	ID d'utilisateur DLS	Objet utilisateur DLS	Renvoyer l'Objet utilisateur DLS en utilisant l'ID d'utilisateur DLS.
GET_NCMT	DLPDU	NCMT	Renvoyer le NCMT de la DLPDU de NCM

7.3 Protocole de gestion DLL

7.3.1 Vue d'ensemble

Le présent Article décrit le protocole d'interface entre la DLM et l'utilisateur DLMS. Cette description du protocole de gestion DLL couvre les services de gestion DLL spécifiés à l'Article 7 en utilisant les services mis à la disposition de l'utilisateur DLMS. Le présent article ne traite pas pleinement les détails de la mise en œuvre et les questions de responsabilité locale.

7.3.2 Définition des primitives

7.3.2.1 Primitives échangées entre DLM et utilisateur DLS

Le Tableau 64 donne un récapitulatif de toutes les primitives échangées entre la DLM et l'utilisateur DLS.

Tableau 64 – Primitives échangées entre DLM et utilisateur DLS

Primitive	Origine	Paramètres associés	Description
DLM-RESET.req	Utilisateur DLS	<Aucun>	Cette primitive de demande entraîne une réinitialisation de la DLE par la DLM.
DLM-RESET.cnf	DLM	Etat DLM	Ceci indique l'état de la réinitialisation.
DLM-SET_VALUE.req	Utilisateur DLS	Nom de la variable Valeur requise	Ce service est utilisé pour attribuer de nouvelles valeurs aux variables de DLE.
DLM-SET_VALUE.cnf	DLM	Etat	L'utilisateur DLMS reçoit une confirmation de l'établissement de nouvelles valeurs aux variables spécifiées.
DLM-GET_VALUE.req	Utilisateur DLS	Nom de la variable	Ce service est utilisé pour lire la valeur d'une variable de DLE.
DLM-GET_VALUE.cnf	DLM	Etat Valeur courante	Ce service renvoie la valeur réelle de la variable spécifiée.
DLM-SAP_ALLOC.req	Utilisateur DLS	SAP ID d'utilisateur DLS	Ce service est utilisé par l'utilisateur DLMS pour obtenir une attribution de SAP de la part de la DLM.
DLM-SAP_ALLOC.cnf	DLM	Etat	Ce service renvoie l'état de résultat de la demande DLM-SAP_ALLOC.req.
DLM-SAP_DEALLOC.req	Utilisateur DLS	SAP	Ce service est utilisé par l'utilisateur DLMS pour libérer et renvoyer au DLM le SAP attribué.
DLM-SAP_DEALLOC.cnf	DLM	Etat	Ce service renvoie l'état de résultat de la demande DLM-SAP_DEALLOC.req.
DLM-GET_SAP_INFO.req	Utilisateur DLS	SAP	Ce service est utilisé par l'utilisateur DLMS pour obtenir de la DLM les informations concernant un SAP déjà attribué.
DLM-GET_SAP_INFO.cnf	DLM	Etat ID d'utilisateur DLS	Ce service renvoie l'état de résultat de la demande DLM-GET_SAP_INFO.req, notamment l'état du résultat et l'ID d'utilisateur DLS.
DLM-GET_DIAG.req	Utilisateur DLS	Type de diag Adresse	Ce service est utilisé par l'utilisateur DLMS pour obtenir de la DLM les informations de diagnostic.
DLM-GET_DIAG.cnf	DLM	Etat Diag	Ce service renvoie l'état de résultat de la demande DLM-GET_SAP_INFO.req, notamment l'état du résultat et les informations de diagnostic.
DLM-EVENT.ind	DLM	Événement	Ce service est utilisé pour informer l'utilisateur DLMS de certains événements ou erreurs dans la DLL.

Les paramètres utilisés avec les primitives échangées entre la DLM et l'utilisateur DLS sont décrits dans le Tableau 65.

Tableau 65 – Paramètres utilisés avec des primitives échangées entre DLM et utilisateur DLS

Paramètre	Description
Etat DLM	Ce paramètre permet à l'utilisateur DLMS de déterminer si le DLMS demandé a été fourni avec succès. En cas d'échec, la raison est spécifiée. Ce paramètre peut prendre l'une des valeurs suivantes: "OK - achevé avec succès" "Echec - achèvement incomplet"
Nom de la variable	Ce paramètre spécifie la variable de DLE dont la valeur doit être établie.
Valeur requise	Ce paramètre spécifie la valeur requise pour la variable sélectionnée.
Etat	Ce paramètre permet à l'utilisateur DLMS de déterminer si le DLMS demandé a été fourni avec succès. En cas d'échec, la raison est spécifiée. Ce paramètre peut prendre l'une des valeurs suivantes: "OK - succès - la variable a pu être mise à jour"; "Défaillance - la variable n'existe pas ou n'a pas pu prendre la nouvelle valeur"; "Echec - paramètres invalides dans la demande"
Valeur courante	Ce paramètre indique la valeur actuelle de la variable désignée.
ID d'utilisateur DLS	Ce paramètre indique l'identification numérique de l'utilisateur DLS local. L'ID d'utilisateur DLS est unique dans un dispositif donné.
SAP	Ce paramètre indique le DLSAP.
Type de diag	Ce paramètre indique le type d'informations de diagnostic. "DIAG_TYPE_L_DEVICE_INFO - informations de dispositif local,"; "DIAG_TYPE_R_DEVICE_INFO - informations de dispositif distant,"; "DIAG_TYPE_NET_INFO - informations de réseau".
Adresse	Ce paramètre indique l'identifiant d'entité DL du nœud désigné.
Événement	Ce paramètre spécifie la primitive ou l'événement composite qui est annoncé. Les valeurs possibles sont définies dans les sections correspondantes de 8.2.

7.3.2.2 Primitives échangées entre DLM et DLPM

Le Tableau 60 donne un récapitulatif de toutes les primitives échangées entre la DLM et la DLPM.

7.3.2.3 Primitives échangées entre DLM et DMAC

Le Tableau 66 et le Tableau 67 résument toutes les primitives et paramètres échangés entre la DLM et le DMAC.

Tableau 66 – Primitives échangées entre DLM et DMAC

Nom de la primitive	Origine	Paramètres associés	Description
MAC-RESET.req	DLM	<Aucun>	Réinitialisation de la couche MAC.
MAC-FW_CTRL.req	DLM	Port R F_en	Ce service est utilisé pour contrôler les fonctions d'acheminement des trames dans la couche MAC.
MAC-FW_CTRL.cnf	D-MAC	Etat	Ce service renvoie l'état de résultat de la demande MAC-FW_CTRL.req.

Tableau 67 – Paramètres utilisés avec des primitives échangées entre DLM et DMAC

Nom du paramètre	Description
Port R	Ce paramètre indique le port Ethernet de Type 21. Chaque dispositif de Type 21 dispose de deux ports R: Port R1 et Port R2.
F_en	Ce paramètre indique l'état du contrôle d'acheminement des trames entre les deux Ports R: "Activer - activer la fonction d'acheminement des trames en fonction du matériel" "Désactiver - désactiver la fonction d'acheminement des trames en fonction du matériel"
Etat	Ce paramètre permet à l'utilisateur DLMS de déterminer si le DLMS demandé a été fourni avec succès. En cas d'échec, la raison est spécifiée. Ce paramètre peut prendre l'une des valeurs suivantes: "OK - succès - la variable a pu être mise à jour"; "Défaillance - la variable n'existe pas ou n'a pas pu prendre la nouvelle valeur"; "Echec - paramètres invalides dans la demande".

7.3.2.4 Primitives échangées entre DLM et DPHY

Le Tableau 68 et le Tableau 69 résument toutes les primitives et paramètres échangés entre la DLM et la DPHY.

Tableau 68 – Primitives échangées entre DLM et DPHY

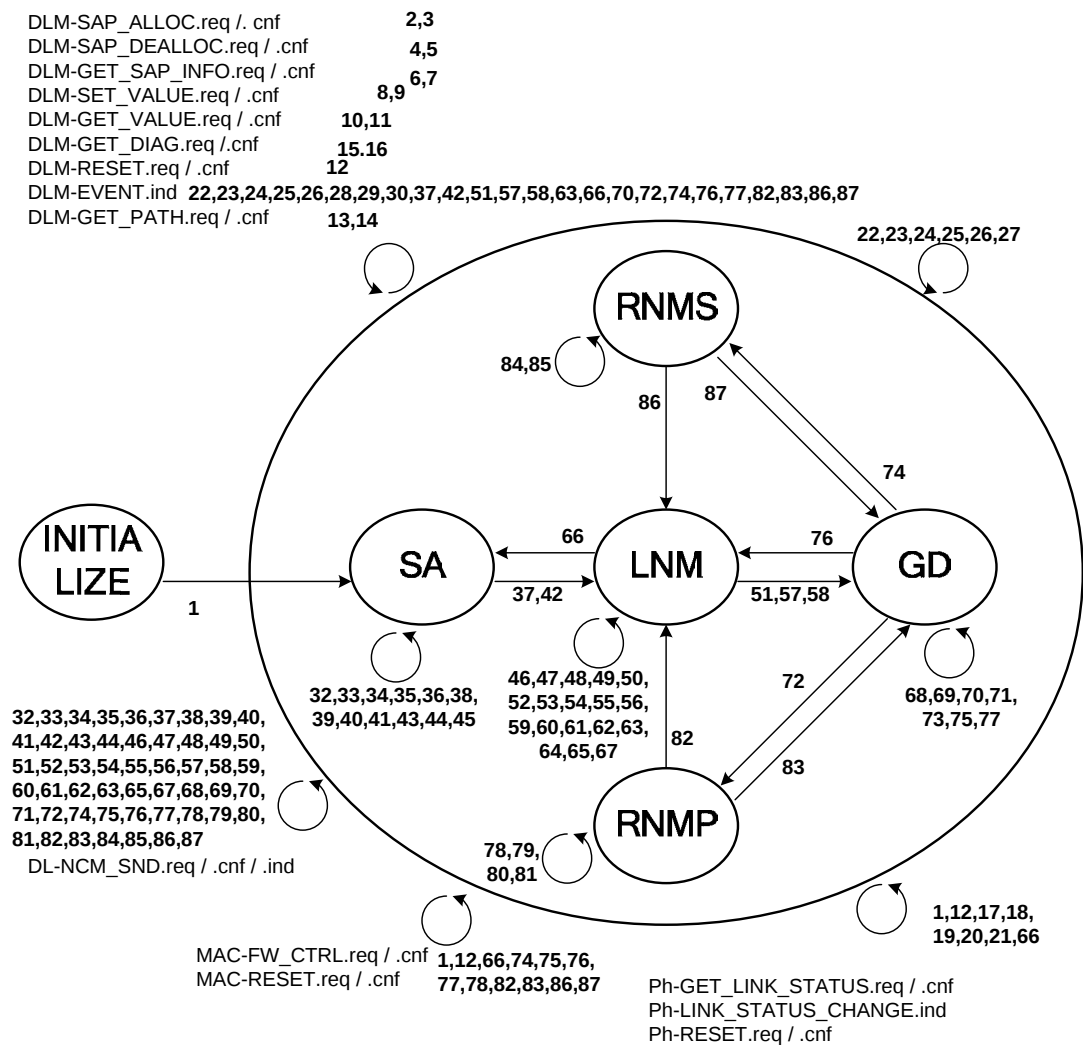
Nom de la primitive	Origine	Paramètres associés	Description
Ph-RESET.req	DLM	<Aucun>	Réinitialisation de la couche physique.
Ph-GET_LINK_STATUS.req	DLM	Port R	Ce service est utilisé pour obtenir de la couche physique des informations d'état de la liaison.
Ph-GET_LINK_STATUS.cnf	DPHY	Port R L_status	Ce service renvoie l'état de résultat de la demande Ph-GET_LINK_STATUS.req
Ph-LINK_STATUS_CHANGE.ind	DPHY	Port R	Ce service est utilisé pour informer la DLM de l'événement de changement d'état de la liaison.

Tableau 69 – Paramètres utilisés avec des primitives échangées entre DLM et DPHY

Nom du paramètre	Description
Port R	Ce paramètre indique le port Ethernet de Type 21. Chaque dispositif de Type 21 dispose de deux ports R: Port R1 et Port R2.
L_status	"liaison active - La liaison est activée et disponible pour communication" "liaison inactive - La liaison n'est pas activée et n'est pas disponible pour communication"

7.3.3 Table d'états DLM

La DLM dispose de 6 états: INITIALISER, SA, LNM, GD, RNMP et RNMS. La DLM contrôle les fonctions d'acheminement des trames dans la couche MAC en fonction de son état DLM. Le diagramme des transitions d'états de la DLM et les descriptions correspondantes sont donnés dans la Figure 22 et dans le Tableau 70.



Légende	
Anglais	Français
Initialize	Initialiser

Figure 22 – Diagramme des transitions d'états de la DLM

SA (autonome, *standalone*)

Cet état signifie que les procédures d'initialisation locale ont été achevées avec succès et que le dispositif est prêt à se connecter aux autres dispositifs. En l'état SA, la DLM tente de trouver les autres dispositifs du réseau. Lorsqu'une liaison est établie, l'état passe à LNM.

LNM (gestionnaire de réseau linéaire)

Cet état signifie que le dispositif local se trouve à l'extrémité d'un réseau linéaire. Le dispositif LNM est connecté au réseau linéaire par l'un de ses deux Ports R. Les deux fonctions d'acheminement des trames sont désactivées dans le dispositif LNM.

GD (dispositif générique)

Cet état signifie que le dispositif local est connecté au réseau par l'intermédiaire de l'un de ses Ports R. Les deux fonctions d'acheminement des trames sont activées dans le dispositif GD. Cependant, dans le dispositif GD, les fonctions d'acheminement des trames sont suspendues jusqu'à ce que le dispositif reçoive DLPDU NCM_LS ou DLPDU NCM_RS.

RNMP (gestionnaire de réseau annulaire primaire)

Cet état signifie que le dispositif local est automatiquement choisi comme étant le gestionnaire de réseau annulaire primaire sur un réseau annulaire. Le dispositif RNMP choisit l'un de ses dispositifs voisins comme gestionnaire de réseau annulaire secondaire (RNMS) en utilisant la DLPDU NCM_RS. Le dispositif RNMP désactive la fonction d'acheminement des trames dans le sens RNMS mais la conserve à l'état activé dans l'autre sens.

RNMS (gestionnaire de réseau annulaire secondaire)

Cet état signifie que le dispositif local est choisi comme étant le gestionnaire de réseau annulaire secondaire sur un réseau annulaire. Le dispositif RNMS désactive la fonction d'acheminement des trames dans le sens RNMP mais la conserve à l'état activé dans l'autre sens.

Tableau 70 – Table d'états DLM

"#	Courant	Événement / Condition =>actions	Etat suivant
1	INITIALIZE	POWER-ON or RESET / => INIT_ENV_VAR() INIT_SAP_INFO() INIT_DEV_INFO() INIT_NET_INFO() INIT_PATH_INFO() MAC-RESET.req { } Ph-RESET.req { } SET_BLOCK_PORT(INVALID_R_PORT) CLEAR_PORT_INFO(R-port1) CLEAR_PORT_INFO(R-port2)	SA
		Les événements déclenchés par le service de gestion DL sont énumérés ci-dessous.	
2	Tout état	DLM-SAP_ALLOC.req {SAP, DLS-user ID} / CHECK_ALLOC_SAP(SAP) = "True" => ALLOC_SAP(SAP, DLS-user ID) Status:= "success" DLM-SAP_ALLOC.cnf {Status}	Tout état
3	Tout état	DLM-SAP_ALLOC.req {SAP, DLS-user ID} / CHECK_ALLOC_SAP(SAP) <> "True" => Status:= "Failure - SAP is already allocated to another DLS-user" DLM-SAP_ALLOC.cnf {Status}	Tout état
4	Tout état	DLM-SAP_DEALLOC.req {SAP} / CHECK_DEALLOC_SAP(SAP) = "True" => DEALLOC_SAP(SAP) Status:= "success" DLM-SAP_DEALLOC.cnf {Status}	Tout état
5	Tout état	DLM-SAP_DEALLOC.req {SAP} / CHECK_DEALLOC_SAP(SAP) <> "True" => Status:= "Failure - SAP is not allocated to a DLS-user" DLM-SAP_DEALLOC.cnf {Status}	Tout état

#	Courant	Événement / Condition =>actions	Etat suivant
6	Tout état	DLM-GET_SAP_INFO.req {SAP} / CHECK_ALLOCEDSAP(SAP) = "True" => DLS-user ID:= GET_USERID_FOR_SAP(SAP) Status:= "success" DLM-GET_SAP_INFO.cnf {Status, DLS-user ID}	Tout état
7	Tout état	DLM-GET_SAP_INFO.req {SAP} / CHECK_ALLOCEDSAP(SAP) <> "True" => DLS-user ID:= INVALID_USER_ID Status:= "Failure - SAP is not allocated to a DLS-user" DLM-GET_SAP_INFO.cnf {Status, DLS-user ID}	Tout état
8	Tout état	DLM-SET_VALUE.req {variable name, desired value} / CHECK_VALUE(variable name, desired value) = "valid" => SET_VALUE(variable name, desired value) Status:= "success" DLM-SET_VALUE.cnf {Status }	Tout état
9	Tout état	DLM-SET_VALUE.req {variable name, desired value} / CHECK_VALUE(variable name, desired value) <> "valid" => Status:= "Failure – invalid parameters in the request" DLM-SET_VALUE.cnf {Status }	Tout état
10	Tout état	DLM-GET_VALUE.req {variable name} / CHECK_VAR(variable name) = "valid" => Current value:= GET_CURRENT_VAL(variable name) Status:= "success" DLM-GET_VALUE.cnf {Status , Current value }	Tout état
11	Tout état	DLM-GET_VALUE.req {variable name} / CHECK_VAR(variable name) <> "valid" => Current value:= INVALID_VALUE Status:= "Failure – invalid parameters in the request" DLM-GET_VALUE.cnf {Status , Current value }	Tout état

"#	Courant	Événement / Condition =>actions	Etat suivant
12	Tout état	DLM-RESET.req { } / => INIT_ENV_VAR() INIT_SAP_INFO() INIT_DEV_INFO() INIT_NET_INFO() INIT_PATH_INFO() MAC-RESET.req { } Ph-RESET.req { } SET_BLOCK_PORT(INVALID_R_PORT) CLEAR_PORT_INFO(R-port1) CLEAR_PORT_INFO(R-port2) Status:= "success" DLM-RESET.cnf {Status}	SA
13	Tout état	DLM-GET_PATH.req {Addr} / CHECK_ADDR(Addr) = "valid" => R-port:= GET_DST_PORT(addr) D_MAC_addr:= GET_DST_MAC_ADDR(addr) Status:= "success" DLM-GET_PATH.cnf {Status, R-port, D_MAC_Addr}	Tout état
14	Tout état	DLM-GET_PATH.req {Addr} / CHECK_ADDR(Addr) <> "valid" => R-port:= INVALID_R_PORT D_MAC_addr:= INVLIID_MAC_ADDR Status:= "Failure – invalid parameters in the request" DLM-GET_PATH.cnf {Status, R-port, D_MAC_Addr}	Tout état
15	Tout état	DLM-GET_DIAG.req {Diag-type, addr} / CHECK_DIAG_TYPE(Diag-type, addr) = "True" => Status:= "success" Diag-info:= GET_DIAG_INFO(Diag-type) DLM-GET_DIAG.cnf {Status, Diag-info}	Tout état
16	Tout état	DLM-GET_DIAG.req {Diag-type, addr} / CHECK_DIAG_TYPE (Diag-type, addr) <> "True" => Status:= "Failure – invalid parameters in the request" DLM-GET_DIAG.cnf {Status, Diag-info}	Tout état
		Les événements déclenchés par le générateur d'événement DLM sont énumérés ci-dessous.	

"#	Courant	Événement / Condition =>actions	Etat suivant
17	Tout état	Ph_LINK_STATUS_CHANGE.ind {R-port} / => Ph-GET_LINK_STATUS.req {R-port}	Tout état
18	Tout état	Ph-GET_LINK_STATUS.cnf {R-port, L_status} / L_status = "link active" && CHECK_NEWLY_LINK_ACTV(R-port) = "True" => NEWLY_LINK_ACTV(R-port)	Tout état
19	Tout état	Ph-GET_LINK_STATUS.cnf {R-port, L_status} / L_status = "link active" && CHECK_NEWLY_LINK_ACTV(R-port) <> "True" => (none)	Tout état
20	Tout état	Ph-GET_LINK_STATUS.cnf {R-port, L_status} / L_status = "link inactive" && CHECK_NEWLY_LINK_INACTV(R-port) = "True" => NEWLY_LINK_INACTV(R-port)	Tout état
21	Tout état	Ph-GET_LINK_STATUS.cnf {R-port, L_status} / L_status = "link inactive" && CHECK_NEWLY_LINK_INACTV(R-port) <> "True" => (none)	Tout état
		Les procédures relatives aux informations de dispositif sont énumérées ci-dessous.	
22	Tout état	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) <> "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) = "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) = "True" => events:= EVENT_THIS_ADDR_COLLISION EVENT_IN_DEVICE DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Tout état
23	Tout état	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) = "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) <> "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) = "True" => events:= EVENT_NET_ADDR_COLLISION EVENT_IN_DEVICE DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Tout état

#	Courant	Événement / Condition =>actions	Etat suivant
24	Tout état	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) = "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) <> "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) <> "True" => events:= EVENT_NET_ADDR_COLLISION DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Tout état
25	Tout état	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) <> "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) = "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) <> "True" => events:= EVENT_THIS_ADDR_COLLISION DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Tout état
26	Tout état	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) <> "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) <> "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) = "True" => events:= EVENT_IN_DEVICE DLM-EVENT.ind(events) SAVE_DEV_INFO(DLMDU, length, R-port)	Tout état
27	Tout état	STORE_DEV_INFO(DLMDU, length, R-port) / CHECK_NET_ADDR_COLLISION(DLMDU) <> "True" && CHECK_THIS_ADDR_COLLISION(DLMDU) <> "True" && CHECK_NEWLY_IN_DEVICE(DLMDU) <> "True" => SAVE_DEV_INFO(DLMDU, length, R-port)	Tout état
		Les procédures relatives à la mise à jour de la table des trajets sont énumérées ci-dessous.	
28	Tout état	DELETE_DEV_INFO(Device UID) / CHECK_UID(UID) = "valid" && CHECK_THIS_ADDR_COLLISION_CLEAR(UID) <> "True" && CHECK_NET_ADDR_COLLISION_CLEAR(UID) <> "True" => DEL_DEV(UID) events:= EVENT_OUT_DEVICE DLM-EVENT.ind(events)	Tout état

#	Courant	Événement / Condition =>actions	Etat suivant
29	Tout état	DELETE_DEV_INFO(Device UID) / CHECK_UID(UID) = "valid" && CHECK_THIS_ADDR_COLLISION_CLEAR(UID) = "True" && CHECK_NET_ADDR_COLLISION_CLEAR(UID) <> "True" => DEL_DEV(UID) events:= EVENT_OUT_DEVICE EVENT_THIS_ADDR_COLLISION_CLEAR DLM-EVENT.ind(events)	Tout état
30	Tout état	DELETE_DEV_INFO(Device UID) / CHECK_UID(UID) = "valid" && CHECK_THIS_ADDR_COLLISION_CLEAR(UID) <> "True" && CHECK_NET_ADDR_COLLISION_CLEAR(UID) = "True" => DEL_DEV(UID) events:= EVENT_OUT_DEVICE EVENT_NET_ADDR_COLLISION_CLEAR DLM-EVENT.ind(events)	Tout état
31	Tout état	DELETE_DEV_INFO(Device UID) / CHECK_UID(UID) <> "valid" => (<none>)	Tout état
		Les transitions d'état sont énumérées ci-dessous.	
32	SA	NEWLY_LINK_ACTV(R-port) / => Block-R-port:= INVERT_PORT(R-port) SET_BLOCK_PORT(Block-R-port) START_TIMER(FamilyReqT) DST_addr:= C_NCM_ADDR NCMT:= NCM_FAMILY_REQ DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
33	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_FAMILY_REQ => SAVE_UID_RP(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length) DST_addr:= C_NCM_ADDR NCMT:= NCM_FAMILY_RES DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA

#	Courant	Événement / Condition =>actions	Etat suivant
34	SA	Timer(FamilyReqT) expired / => START_TIMER(FamilyReqT) DST_addr:= C_NCM_ADDR NCMT:= NCM_FAMILY_REQ DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
35	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_FAMILY_RES => STOP_TIMER(FamilyReqT) SAVE_UID_RP(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM_FAMILY) START_TIMER(MediaLinkedT) DST_addr:= C_NCM_ADDR NCMT:= NCM_MEDIA_LINKED DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
36	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "False" => (<none>)	SA

#	Courant	Événement / Condition =>actions	Etat suivant
37	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ML) = "True" => SET_BLOCK_PORT(INVALID_R_PORT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) events:= EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr:= C_NCM_ADDR NCMT:= NCM_ADV_THIS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} DST_addr:= C_NCM_ADDR NCMT:= NCM_LINE_START DLMDU:= Informations de dispositif local & informations de réseau length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNM
38	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ML) = "False" => STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ADV) DST_addr:= C_NCM_ADDR NCMT:= NCM_ADV_THIS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA

#	Courant	Événement / Condition =>actions	Etat suivant
39	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "False" => STORE_DEV_INFO(DLMDU, Length) DST_addr:= C_NCM_ADDR NCMT:= NCM_ADV_THIS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
40	SA	Timer(MediaLinkedT) expired / => START_TIMER(MediaLinkedT) DST_addr:= C_NCM_ADDR NCMT:= NCM_MEDIA_LINKED DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	SA
41	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "False" => (<none>)	SA

#	Courant	Événement / Condition =>actions	Etat suivant
42	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "True" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "True" CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "True") => STOP_TIMER(MediaLinkedT) SET_BLOCK_PORT(INVALID_R_PORT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) events:= EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr:= C_NCM_ADDR NCMT:= NCM_LINE_START DLMDU:= Informations de dispositif local & informations de réseau length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNM
43	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "False" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "True" CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "True") => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ML)	SA
44	SA	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "False" && CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "False") => STORE_DEV_INFO(DLMDU, Length)	SA

#	Courant	Événement / Condition =>actions	Etat suivant
45	SA	NEWLY_LINK_INACTV(R-port) / => INIT_DEV_INFO() INIT_NET_INFO() INIT_PATH_INFO() SET_BLOCK_PORT(INVALID_R_PORT) CLEAR_PORT_INFO(R-port1) CLEAR_PORT_INFO(R-port2)	SA
46	LNМ	NEWLY_LINK_ACTV(R-port) / => START_TIMER(FamilyReqT) DST_addr:= C_NCM_ADDR NCMT:= NCM_FAMILY_REQ DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
47	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_FAMILY_REQ => SAVE_UID_RP(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length) DST_addr:= C_NCM_ADDR NCMT:= NCM_FAMILY_RES DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
48	LNМ	Timer(FamilyReqT) expired / => START_TIMER(FamilyReqT) DST_addr:= C_NCM_ADDR NCMT:= NCM_FAMILY_REQ DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ

#	Courant	Événement / Condition =>actions	Etat suivant
49	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_FAMILY_RES => STOP_TIMER(FamilyReqT) SAVE_UID_RP(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM_FAMILY) START_TIMER(MediaLinkedT) DST_addr:= C_NCM_ADDR NCMT:= NCM_MEDIA_LINKED DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNM
50	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "False" => (<none>)	LNM
51	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ML) = "True" => STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHAGNE_DLM_STATE(GD) events:= EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) CLEAR_LNM_UID(R-port) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr:= C_NCM_ADDR NCMT:= NCM_ADV_THIS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	GD

#	Courant	Événement / Condition =>actions	Etat suivant
52	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "True" && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ML) = "False" => STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ADV) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr:= C_NCM_ADDR NCMT:= NCM_ADV_THIS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNM
53	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_UID_RP(DLMDU, R-port) = "False" && CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr:= C_NCM_ADDR NCMT:= NCM_ADV_THIS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNM
54	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_ECHO(DLMDU, Length) = "True" => (<none>)	LNM

#	Courant	Événement / Condition =>actions	Etat suivant
55	LNМ	Timer(MediaLinkedT) expired / => START_TIMER(MediaLinkedT) DST_addr:= C_NCM_ADDR NCMT:= NCM_MEDIA_LINKED DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNМ
56	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "False" => (<none>)	LNМ
57	LNМ	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "True" && CHECK_LNM_UID(DLMDU) = "True" && CHECK_UID_RP(DLMDU, R-port) = "True" => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHAGNE_DLM_STATE(GD) events:= EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) CLEAR_LNM_UID(R-port) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	GD

#	Courant	Événement / Condition =>actions	Etat suivant
58	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "True" && CHECK_LNM_UID(DLMDU) = "False" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "True" CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "True") => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_CFM) CHAGNE_DLM_STATE(GD) events:= EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) CLEAR_LNM_UID(R-port) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	GD
59	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "False" && CHECK_LNM_UID(DLMDU) = "True" && CHECK_UID_RP(DLMDU, R-port) = "True" => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ML) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	LNM

#	Courant	Événement / Condition =>actions	Etat suivant
60	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_PORT_INFO(R-port, PORT_WAIT_ADV) = "False" && CHECK_LNM_UID(DLMDU) = "False" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "True" CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "True") => STOP_TIMER(MediaLinkedT) STORE_DEV_INFO(DLMDU, Length) UPDATE_PORT_INFO(R-port, PORT_WAIT_ML) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	LNM
61	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_LNM_UID(DLMDU) = "True" && CHECK_UID_RP(DLMDU, R-port) = "False" && CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	LNM
62	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_LNM_UID(DLMDU) = "False" && (CHECK_NET_PORT_INFO(DLMDU, PORT1_LINK_DOWN) = "False" && CHECK_NET_PORT_INFO(DLMDU, PORT2_LINK_DOWN) = "False") CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	LNM

#	Courant	Événement / Condition =>actions	Etat suivant
63	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_PORT_INFO(R-port, PORT_CFM_FAMILY) = "True" && CHECK_ECHO(DLMDU, Length) = "True" => CHANGE_TOPOLOGY(NET_TPG_RING) events:= EVENT_NET_TPG_CHG DLM-EVENT.ind(events) START_TIMER(ChangeRingStateT)	LNM
64	LNM	Timer(ChangeRingStateT) expired / => START_TIMER(ChangeRingStateT)	LNM
65	LNM	NEWLY_LINK_INACTV(R-port) / CHECK_PORT_INFO(R-port, PORT_CFM) = "False" => DELETE_PATH_INFO(R-port) CLEAR_PORT_INFO(R-port) DST_addr:= C_NCM_ADDR NCMT:= NCM_LINE_START DLMDU:= Informations de dispositif local & informations de réseau length:= Taille de DLMDU R-port:= INVERT_PORT(R-port) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	LNM
66	LNM	NEWLY_LINK_INACTV(R-port) / CHECK_PORT_INFO(R-port, PORT_CFM) = "True" => CHANGE_TOPOLOGY(NET_TPG_SA) CHAGNE_DLM_STATE(SA) events:= EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) UPDATE_PATH_TABLE(SA) INIT_DEV_INFO() INIT_NET_INFO() SET_BLOCK_PORT(INVALID_R_PORT) CLEAR_PORT_INFO(R-port1) CLEAR_PORT_INFO(R-port2) MAC-RESET.req { } Ph-RESET.req { }	SA

#	Courant	Événement / Condition =>actions	Etat suivant
67	LNM	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_LINE_START => CHANGE_LNM_UID(DLMDU, R-port) STORE_DEV_INFO(DLMDU, Length)	LNM
68	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_ECHO(DLMDU, Length) = "True" => (<none>)	GD
69	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_MEDIA_LINKED && CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr:= C_NCM_ADDR NCMT:= NCM_ADV_THIS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	GD
70	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_ECHO(DLMDU, Length) = "True" => CHANGE_TOPOLOGY(NET_TPG_RING) events:= EVENT_NET_TPG_CHG DLM-EVENT.ind(events) START_TIMER(ChangeRingStateT)	GD
71	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ADV_THIS && CHECK_ECHO(DLMDU, Length) = "False" => STORE_DEV_INFO(DLMDU, Length) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port}	GD

#	Courant	Événement / Condition =>actions	Etat suivant
72	GD	Timer(ChangeRingStateT) expired / CHECK_NET_TOPOLOGY() = NET_TPG_RING && CHECK_RNMP() = RNMP => STOP_TIMER(ChangeRingStateT) CHAGNE_DLM_STATE(RNMP) events:= EVENT_DEV_STATE_CHG DLM-EVENT.ind (events) CLEAR_LNM_UID(R-port1) CLEAR_LNM_UID(R-port2) DST_addr:= C_NCM_ADDR NCMT:= NCM_RING_START DLMDU:= Informations de dispositif local & informations de réseau length:= Taille de DLMDU Rnms_UID:= GET_RNMS_UID() SET_RNMPS_UID(Rnms_UID, DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port2} START_TIMER(AckRNMT)	RNMP
73	GD	Timer(ChangeRingStateT) expired / CHECK_NET_TOPOLOGY() = NET_TPG_RING && CHECK_RNMP() <> RNMP => STOP_TIMER(ChangeRingStateT)	GD

#	Courant	Événement / Condition =>actions	Etat suivant
74	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RING_START && GET_NET_RNMS_UID(DLMDU) = device information.UID => STORE_DEV_INFO(DLMDU, Length) CHAGNE_DLM_STATE(RNMS) events:= EVENT_DEV_STATE_CHG DLM-EVENT.ind (events) CLEAR_LNM_UID(R-port1) CLEAR_LNM_UID(R-port2) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr:= SRC_addr NCMT:= NCM_ACK_RNMS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} Forward-control:= FW_DISABLE R-port:= GET_R_PORT_FOR(addr of RNMP) FW_CTRL(R-port, Forward-control) R-port:= INVERT_PORT(GET_R_PORT_FOR(addr of RNMP)) Forward-control:= FW_ENABLE FW_CTRL(R-port, Forward-control)	RNMS
75	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RING_START && GET_NET_RNMS_UID(DLMDU) <> device information.UID => STORE_DEV_INFO(DLMDU, Length) CLEAR_LNM_UID(R-port1) CLEAR_LNM_UID(R-port2) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} Forward_control:= FW_ENABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	GD

#	Courant	Événement / Condition =>actions	Etat suivant
76	GD	NEWLY_LINK_INACTV(R-port) / => DELETE_PATH_INFO(R-port) CLEAR_PORT_INFO(R-port) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) events:= EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr:= C_NCM_ADDR NCMT:= NCM_LINE_START DLMDU:= Informations de dispositif local & informations de réseau length:= Taille de DLMDU R-port:= INVERT_PORT(R-port) DL-NCM_SND.req(DST_addr, NCMT, DLMDU, Length, R-port) UPDATE_PATH_TABLE(LINE) Forward_control:= FW_DISABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	LNM
77	GD	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_LINE_START => STORE_DEV_INFO(DLMDU, Length) CHANGE_TOPOLOGY(NET_TPG_LINE) events:= EVENT_NET_TPG_CHG DLM-EVENT.ind (events) UPDATE_PATH_TABLE(LINE) CHANGE_LNM_UID(DLMDU, R-port) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} Forward_control:= FW_ENABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	GD

#	Courant	Événement / Condition =>actions	Etat suivant
78	RNMP	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ACK_RNMS && CHECK_RNMS_UID(DLMDU) = "True" => STOP_TIMER(AckRNMST) Forward-control:= FW_DISABLE R-port:= GET_R_PORT_FOR(addr of RNMS) FW_CTRL(R-port, Forward-control) R-port:= INVERT_PORT(GET_R_PORT_FOR(addr of RNMS)) Forward-control:= FW_ENABLE FW_CTRL(R-port, Forward-control)	RNMP
79	RNMP	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_ACK_RNMS && CHECK_RNMS_UID(DLMDU) = "False" => (<none>)	RNMP
80	RNMP	Timer(AckRNMST) expired / => DST_addr:= addr of RNMS NCMT:= NCM_RETRY_RNMS DLMDU:= Informations de dispositif local length:= Taille de DLMDU R-port:= both R-port DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port} START_TIMER(AckRNMST)	RNMP
81	RNMP	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RING_START && CHECK_ECHO(DLMDU, Length) = "True" => (<none>)	RNMP

#	Courant	Événement / Condition =>actions	Etat suivant
82	RNMP	NEWLY_LINK_INACTV(R-port) / => DELETE_PATH_INFO(R-port) CLEAR_PORT_INFO(R-port) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) events:= EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr:= C_NCM_ADDR NCMT:= NCM_LINE_START DLMDU:= Informations de dispositif local & informations de réseau length:= Taille de DLMDU R-port:= INVERT_PORT(R-port) DL-NCM_SND.req(DST_addr, NCMT, DLMDU, Length, R-port) UPDATE_PATH_TABLE(LINE) Forward_control:= FW_DISABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	LNM
83	RNMP	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_LINE_START => STORE_DEV_INFO(DLMDU, Length) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(GD) events:= EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) UPDATE_PATH_TABLE(LINE) CHANGE_LNM_UID(DLMDU, R-port) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} Forward_control:= FW_ENABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	GD

#	Courant	Événement / Condition =>actions	Etat suivant
84	RNMS	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RING_START && GET_NET_RNMS_UID(DLMDU) = device information.UID => STORE_DEV_INFO(DLMDU, Length) CLEAR_LNM_UID(R-port1) CLEAR_LNM_UID(R-port2) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} DST_addr:= SRC_addr NCMT:= NCM_ACK_RNMS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	RNMS
85	RNMS	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_RETRY_RNMS => STORE_DEV_INFO(DLMDU, Length) DST_addr:= SRC_addr NCMT:= NCM_ACK_RNMS DLMDU:= Informations de dispositif local length:= Taille de DLMDU DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, R-port}	RNMS
86	RNMS	NEWLY_LINK_INACTV(R-port) / => DELETE_PATH_INFO(R-port) CLEAR_PORT_INFO(R-port) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(LNM) events:= EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) DST_addr:= C_NCM_ADDR NCMT:= NCM_LINE_START DLMDU:= Informations de dispositif local & informations de réseau length:= Taille de DLMDU R-port:= INVERT_PORT(R-port) DL-NCM_SND.req(DST_addr, NCMT, DLMDU, Length, R-port) UPDATE_PATH_TABLE(LINE) Forward_control:= FW_DISABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	LNM

#	Courant	Événement / Condition =>actions	Etat suivant
87	RNMS	DL-NCM_SND.ind {DST_addr, SRC_addr, NCMT, DLMDU, Length, R-port} / NCMT = NCM_LINE_START => STORE_DEV_INFO(DLMDU, Length) CHANGE_TOPOLOGY(NET_TPG_LINE) CHAGNE_DLM_STATE(GD) events:= EVENT_NET_TPG_CHG EVENT_DEV_STATE_CHG DLM-EVENT.ind(events) UPDATE_PATH_TABLE(LINE) CHANGE_LNM_UID(DLMDU, R-port) Forward-R-port:= INVERT_PORT(R-port) INC_HOP_CNT(DLMDU) DL-NCM_SND.req {DST_addr, NCMT, DLMDU, Length, Forward-R-port} Forward_control:= FW_ENABLE FW_CTRL(R-port1, Forward_control) FW_CTRL(R-port2, Forward_control)	GD

7.3.4 Fonctions de DLM

Le Tableau 71 présente les fonctions internes assurées par la DLM.

Tableau 71 – Table des fonctions DLM

Nom de la fonction	Entrée	Sortie	Fonctionnement
INIT_ENV_VAR			Initialiser paramètres de configuration DLL (voir 4.6.2).
INIT_SAP_INFO			Initialiser informations SAP (voir 4.6.4).
INIT_DEV_INFO			Initialiser informations de dispositif local (voir 4.6.5).
INIT_NET_INFO			Initialiser informations de réseau (voir 4.6.6).
INIT_PATH_INFO			Initialiser informations de table des trajets (voir 4.6.7).
CHECK_ALLOC_SAP	SAP	Vrai/Faux	Renvoyer "VRAI" si le SAP est disponible. Sinon, renvoyer "FAUX".
ALLOC_SAP	SAP ID d'utilisateur DLS		Enregistrer ID d'utilisateur DLS dans les informations SAP.
CHECK_DEALLOC_SAP	SAP	Vrai/Faux	Vérifier si l'attribution du SAP peut être retirée.
DEALLOC_SAP	SAP		Retirer l'ID d'utilisateur DLS des informations SAP.
CHECK_ALLOCEDSAP	SAP	Vrai/Faux	Vérifier si le SAP est déjà attribué à un quelconque utilisateur DLS.
GET_USERID_FOR_SAP	SAP	ID d'utilisateur DLS	Renvoyer l'ID d'utilisateur DLS auquel est attribué le SAP.
CHECK_VALUE	Nom de la variable Valeur requise	Valide/Non valide	Vérifier que les variables demandées avec la valeur requise sont valides.
SET_VALUE	Nom de la variable Valeur requise		Etablir la valeur de la variable demandée.
GET_CURRENT_VAL	Nom de la variable	Current_value	Obtenir la valeur de la variable demandée.
CHECK_ADDR	Identifiant d'entité DL	Valide/Non valide	Vérifier si l'identifiant d'entité DL désigné est valide. Renvoyer "Valide" si l'identifiant d'entité DL monodestinataire est enregistré dans la table des trajets. Renvoyer "Non valide" si l'identifiant d'entité DL monodestinataire n'est pas enregistré dans la table des trajets. Renvoyer "Valide" si l'identifiant d'entité DL désigné n'est pas dans la plage de l'identifiant d'entité DL monodestinataire.
GET_DST_PORT	Identifiant d'entité DL	Port R	Renvoyer le Port R de destination pour le dispositif désigné dans la table des trajets.
GET_DST_MAC_ADDR	Adresse	Adresse MAC	Renvoyer l'adresse MAC de l'ISO/CEI 8802:3:2000 pour le dispositif désigné dans la table des trajets.
CHECK_DIAG_TYPE	Diagnostic_type Adresse	Vrai/Faux	Vérifier si le type d'informations de diagnostic provenant de l'utilisateur DLS est disponible.
GET_DIAG_INFO	Diagnostic_type Adresse	Informations de diagnostic	Renvoyer les informations de diagnostic selon le type d'informations de diagnostic.
CHECK_NEWLY_LINK_A CTV	Port R	Vrai/Faux	Vérifier si le Port R est passé de l'état "liaison Inactive" à "liaison Active".

Nom de la fonction	Entrée	Sortie	Fonctionnement
CHECK_NEWLY_LINK_I NACTV	Port R	Vrai/Faux	Vérifier si le Port R est passé de l'état "liaison Active" à "liaison Inactive".
UPDATE_PORT_INFO	Port R état		Mettre à jour les informations de port dans les informations du dispositif (voir 4.6.5.9).
CLEAR_PORT_INFO	Port R		Effacer les informations de port pour le Port R.
CHECK_PORT_INFO	Port R état	Vrai/Faux	Vérifier si le bit d'information de port pour le Port R dans les informations du dispositif est établi. Renvoyer "VRAI" si le bit d'état dans les informations de port pour le Port R est établi. Sinon, renvoyer "FAUX".
SET_BLOCK_PORT	Port R		Etablir le Port R à l'état bloqué. Etablir Port R1 et Port R2 à l'état actif si Port R est INVALID_R_PORT. Sinon, mettre Port R à l'état bloqué.
CHECK_ECHO	DLMDU longueur	Vrai/Faux	Vérifier si la DLPDU a été générée par le dispositif lui-même.
INC_HOP_CNT	DLMDU		Incrémenter le champ Nombre de sauts dans la DLMDU.
INVERT_PORT	Port R	Port R	Inverser l'interface logique vers le Port R. Renvoyer Port R2 si l'entrée est Port R1 et retourner Port R1 si l'entrée est Port R2.
CHECK_NET_TOPOLOGY		Topologie du réseau	Vérifier si le réseau est configuré en topologie annulaire. Renvoyer NET_TPG_RING si le Message de contrôle du réseau diffusé est reçu par le dispositif lui-même. Sinon, retourner NET_TPG_LINE.
CHECK_NET_ADDR_COLLISION	DLMDU	Vrai/Faux	Renvoyer "VRAI" si l'identifiant d'entité DL d'origine dans la DLMDU est dupliqué à l'un des deux autres dispositifs sur le réseau. Sinon, renvoyer "FAUX".
CHECK_THIS_ADDR_COLLISION	DLMDU		Renvoyer "VRAI" si l'identifiant d'entité DL d'origine dans la DLMDU est dupliqué dans le dispositif local lui-même. Sinon, renvoyer "FAUX".
CHECK_NEWLY_IN_DEVICE	DLMDU	Vrai/Faux	Vérifier si l'identifiant d'entité DL d'origine dans la DLMDU n'est pas enregistré dans les informations de Dispositif.
SAVE_DEV_INFO	DLMDU longueur Port R		Enregistrer les informations de dispositif de la DLMDU vers la table des trajets.
DEL_DEV	UID de dispositif		Retirer de la table des trajets l'élément correspondant à l'UID de dispositif.
DELETE_PATH_INFO	Port R		Retirer de la table des trajets l'élément correspondant au Port R.
CHANGE_TOPOLOGY	Topologie		Mettre à jour la Topologie dans les informations de réseau.
CHANGE_LNM_UID	DLMDU Port R		Enregistrer l'identifiant d'entité DL d'origine dans la DLMDU vers l'UID de dispositif LNM pour le Port R dans les informations de réseau.
CLEAR_LNM_UID	Port R		Eliminer l'UID de dispositif LNM pour le Port R dans les informations de réseau.
CHANGE_DLM_STATE	Etat DLM		Mettre à jour l'état DLS dans les informations du dispositif.

Nom de la fonction	Entrée	Sortie	Fonctionnement
SAVE_UID_RP	DLMDU Port R		Enregistrer l'identifiant d'entité DL d'origine dans la DLMDU vers l'UID de Dispositif pour le Port R dans les informations du dispositif.
CHECK_UID_RP	DLMDU Port R	Vrai/Faux	Renvoyer "VRAI" si l'identifiant d'entité DL d'origine dans la DLMDU a été enregistré comme UID de Dispositif pour le Port R dans les informations de dispositif. Sinon, renvoyer "FAUX".
CHECK_NET_PORT_INF O	DLMDU état	Vrai/Faux	Vérifier si le bit d'information de port dans la DLMDU est établi. Renvoyer "VRAI" si le bit d'état dans les informations de port de la DLMDU est établi. Sinon, renvoyer "FAUX".
CHECK_LNM_UID	DLMDU	Vrai/Faux	Renvoyer "VRAI" si l'identifiant d'entité DL d'origine dans la DLMDU a été enregistré comme LNM dans les informations de réseau. Sinon, renvoyer "FAUX".
UPDATE_PATH_TABLE	Topologie		Mettre à jour et recalculer le port de destination dans la table des trajets selon la topologie du réseau.
CHECK_LNMS		Vrai/Faux	Vérifier si deux LNM sont inscrites dans la table des trajets.
FW_CTRL	Port R Contrôle		Activer ou désactiver la fonction d'acheminement des trames pour le Port R.
GET_RNMS_UID		UID	Renvoyer l'UID de dispositif enregistré comme RNMS dans la table des trajets. Renvoyer INVALID_UID si le RNMS ne se trouve pas dans la table des trajets.
GET_NET_RNMS_UID	DLMDU	UID	Renvoyer l'UID de dispositif RNMS dans la DLMDU.
CHECK_RNMS_UID	DLMDU	Vrai/Faux	Renvoyer "VRAI" si l'identifiant d'entité DL d'origine dans la DLMDU est identique à l'UID de dispositif RNMS dans les informations de réseau. Sinon, renvoyer "FAUX".
GET_R_PORT_FOR	Identifiant d'entité DL	Port R	Renvoyer le Port R de destination vers l'identifiant d'entité DL de la table des trajets. Renvoyer INVALID_ADDR s'il n'est trouvé aucune entrée dans la table.
START_TIMER	temps requis		Démarrer le temporisateur.
STOP_TIMER	temps requis		Arrêter le temporisateur.

8 Constantes et codes d'erreur

8.1 Généralités

Le présent Article décrit les constantes de DLL de Type 21 et leurs valeurs.

8.2 Constantes

Le Tableau 72 résume l'ensemble des constantes de DLL.

Tableau 72 – Constantes de DLL

Constante	Valeur	Description
C_HIGEST_PRIORITY	3	La priorité de message la plus élevée
INVALID_USER_ID	0	Identification d'utilisateur non valide
INVALID_R_PORT	0	Port R non valide
NCM_FAMILY_REQ	1	Message de contrôle du réseau "link active req." (demande d'activation de liaison)
NCM_FAMILY_RES	2	Message de contrôle du réseau "link active res." (réponse d'activation de liaison)
NCM_MEDIA_LINKED	3	Message de contrôle du réseau "indicate established Type 21 link" (indication de liaison de Type 21 établie)
NCM_ADV_THIS	4	Message de contrôle du réseau "advertise this information" (annoncer cette information)
NCM_LINE_START	5	Message de contrôle du réseau "starting line topology network" (début de réseau en topologie linéaire)
NCM_RING_START	6	Message de contrôle du réseau "starting ring topology network" (début de réseau en topologie annulaire)
NCM_ACK_RNMS	7	Message ACK (acquiescement) de RNMS vers RNMP
NCM_RETRY_RNMS	8	Message REQ (demande) de RNMP vers RNMS
EVENT_NET_TPG_CHG	0x01	Événement généré lorsque la topologie du réseau est modifiée
EVENT_DEV_STATE_CHG	0x02	Événement généré lorsque l'état DLM est modifié
EVENT_THIS_ADDR_COLLISION	0x04	Événement généré lorsque la collision d'identifiant d'entité DL locale est détectée
EVENT_THIS_ADDR_COLLISION_CLEAR	0x08	Événement généré lorsque la collision d'identifiant d'entité DL locale est éliminée
EVENT_NET_ADDR_COLLISION	0x10	Événement généré lorsque la collision d'identifiant d'entité DL du réseau est détectée
EVENT_NET_ADDR_COLLISION_CLEAR	0x20	Événement généré lorsque la collision d'identifiant d'entité DL du réseau est éliminée
EVENT_IN_DEVICE	0x40	Événement généré lorsqu'un dispositif rejoint le réseau pour la première fois
EVENT_OUT_DEVICE	0x80	Événement généré lorsqu'un dispositif est déconnecté du réseau
C_NCM_ADDR	0xFFFE	L'identifiant d'entité DL pour des messages de contrôle du réseau
C_BROADCAST_ADDR	0xFFFF	Identifiant d'entité DL pour la diffusion
FW_ENABLE	1	Activer la fonction d'acheminement des trames
FW_DISABLE	0	Désactiver la fonction d'acheminement des trames
INVALID_UID	0	Identifiant unique non valide
INVALID_ADDR	0xFFDC	adresse non valide
PORT_LINK_DOWN	0x01	Liaison Port R hors-service
PORT_CFM_FAMILY	0x02	Trame NCM_FAMILY_RES reçue du port R
PORT_WAIT_ADV	0x04	Attente de la trame NCM_ADV_THIS du port R
PORT_WAIT_ML	0x08	Attente de la trame NCM_MEDIA_LINKED du port R
PORT_CFM	0x10	Confirmation d'état du dispositif
PORT1_LINK_DOWN	0x0001	Liaison Port R1 hors-service
PORT2_LINK_DOWN	0x0100	Liaison Port R2 hors-service
FamilyReqT	100 ms	Temps d'attente de trame NCM_FAMILY_RES

Constante	Valeur	Description
MediaLinkedT	100 ms	Temps d'attente de trame NCM_ADV_THIS
AckRNMST	100 ms	Temps d'attente de trame NCM_ACK_RNMS
ChangeRingStateT	1 ms	Temps imparti pour le changement d'état de l'anneau

8.3 Codes d'erreur de la couche Liaison de données

Le Tableau 73 résume l'ensemble des codes d'erreur de DLL.

Tableau 73 – Codes d'erreur de DLL de Type 21

Erreur	Valeur	Description
ERR_FRAME_SIZE	0x1001	Erreur de taille de trame
ERR_PROTOCOL_VER	0x1002	Erreur de version de protocole
ERR_NOT4ME_FRAME	0x1003	La trame reçue n'est pas destinée au dispositif local
ERR_SAP	0x1004	Erreur de point d'accès de service
ERR_SERVICE_CMD	0x1005	Erreur de commande de service
ERR_NO_DST_ADDR	0x1006	Aucun dispositif trouvé ne correspond à l'identifiant d'entité DL de destination
ERR_PRM	0x1007	Paramètre non valide
ERR_NG_DEV_INFO	0x1008	Erreur d'informations de dispositif
ERR_GET_DEV_INFO	0x1009	Erreur de taille d'informations de dispositif
ERR_Q_ALREADY_FULL	0x100A	File d'attente pleine
ERR_SAP_ALREADY_REGED	0x100B	Le SAP est déjà attribué à un utilisateur DLS
ERR_SAP_NOT_REGED	0x100C	Le SAP n'est pas attribué à un utilisateur DLS
ERR_NO_MATCHED_ITEM	0x100D	Aucun élément trouvé dans la NMIB.
ERR_INVALID_DIAG_TYPE	0x100E	Le type d'informations de diagnostic n'est pas valide
ERR_DEV_CNT_EXIED	0x2001	L'identifiant d'entité DL est supérieur à MAX_ADDR.
NOT_SUPPORTEDSAP	0x2002	Erreur d'adresse SAP non valide
INVALID_DEV_INFO	0x2003	Erreur d'informations de dispositif non valides

Bibliographie

IEC/TR 61158-1:2010³, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the CEI 61158 and CEI 61784 series* (disponible uniquement en anglais)⁴

IEC 61158-2:2010³, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition* (disponible uniquement en anglais)

IEC 61158-5-21:2010³, *Industrial communication networks – Fieldbus specifications – Part 5-21: Application layer service definition – Type 21 elements* (disponible uniquement en anglais)

IEC 61158-6-21:2010³, *Industrial communication networks – Fieldbus specifications – Part 6-21: Application layer protocol specification – Type 21 elements* (disponible uniquement en anglais)

IEC 61588, *Precision clock synchronization protocol for networked measurement and control systems* (disponible uniquement en anglais)

ISO/CEI TR8802-1, *Technologies de l'information – Télécommunications et échange d'information entre systèmes – Réseaux locaux et métropolitains – Exigences spécifiques – Partie 1 : Vue d'ensemble des normes de réseaux locaux*

IETF RFC 768, *User Datagram Protocol (UDP)*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 791, *Internet Protocol*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 793, *Transmission Control Protocol*, disponible à l'adresse <<http://www.ietf.org>>

³ A publier.

⁴ Les publications monolingues des séries IEC 61158 et IEC 61784 sont actuellement en cours de traduction.

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

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