

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 4-25: Data-link layer protocol specification – Type 25 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 4-25: Spécification du protocole de la couche liaison de données –
Éléments de type 25**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2019 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 l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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

Tel.: +41 22 919 02 11
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 corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables 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 It also gives information on projects, replaced and withdrawn publications.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 online and once a month by email.

IEC 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: sales@iec.ch.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) 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 IEC

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

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC 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.

IEC Just Published - webstore.iec.ch/justpublished

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

Service Clients - webstore.iec.ch/csc

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

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

67 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.



IEC 61158-4-25

Edition 1.0 2019-04

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 4-25: Data-link layer protocol specification – Type 25 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 4-25: Spécification du protocole de la couche liaison de données –
Éléments de type 25**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.20; 35.110

ISBN 978-2-8322-9142-9

**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..... | 5 |
| INTRODUCTION..... | 7 |
| 1 Scope..... | 8 |
| 1.1 General..... | 8 |
| 1.2 Specifications | 8 |
| 1.3 Procedures | 8 |
| 1.4 Applicability | 9 |
| 1.5 Conformance | 9 |
| 2 Normative references | 9 |
| 3 Terms, definitions, symbols, abbreviations and conventions | 9 |
| 3.1 Reference model terms and definitions | 10 |
| 3.2 Service convention terms and definitions | 11 |
| 3.3 Terms and definitions..... | 11 |
| 3.4 Symbols and abbreviations | 13 |
| 3.5 Common conventions..... | 14 |
| 3.6 Additional Type 25 conventions | 16 |
| 3.6.1 Primitive conventions..... | 16 |
| 3.6.2 State machine conventions..... | 16 |
| 4 Overview of the DL-protocol | 17 |
| 4.1 General..... | 17 |
| 4.2 Overview of the medium access control | 17 |
| 4.2.1 General | 17 |
| 4.2.2 Network topology..... | 18 |
| 4.2.3 Priority control with VLAN..... | 19 |
| 4.2.4 The maximum delivery delay in Type 25 network | 20 |
| 4.2.5 Traffic control for real-time communication | 21 |
| 4.3 Service assumed from PhL | 21 |
| 4.4 DL Layer architecture..... | 22 |
| 4.5 Local parameters and variables | 23 |
| 4.5.1 Overview | 23 |
| 4.5.2 Variables, parameter, counter and timer | 23 |
| 5 General structure and encoding of PhPDUs and DLPDU and related elements of procedure..... | 24 |
| 5.1 Overview | 24 |
| 5.2 Common MAC frame structure, encoding and elements of procedure..... | 24 |
| 5.2.1 MAC frame structure..... | 24 |
| 5.2.2 Elements of the MAC frame | 25 |
| 6 DLPDU-specific structure, encoding and elements of procedure | 27 |
| 6.1 General..... | 27 |
| 6.2 Structure of the RCL DLPDU..... | 27 |
| 6.2.1 RCL header | 27 |
| 7 DLE elements of procedure | 29 |
| 7.1 Overview | 29 |
| 7.2 RCL communication control (RCLC)..... | 29 |
| 7.2.1 General | 29 |
| 7.2.2 Primitive definitions | 29 |

| | | |
|-------|--|----|
| 7.2.3 | RCLC state machine..... | 32 |
| 7.2.4 | Function of RCLC | 48 |
| 7.3 | Real-time communication control (RTC)..... | 48 |
| 7.3.1 | General | 48 |
| 7.3.2 | Primitive definitions | 48 |
| 7.3.3 | RTC state machine | 50 |
| 7.3.4 | Function of RTC | 51 |
| 7.4 | Transmit/Receive control (TRC)..... | 52 |
| 7.4.1 | General | 52 |
| 7.4.2 | Primitive definitions | 52 |
| 7.4.3 | TRC state machine | 52 |
| 7.4.4 | Function of TRC | 57 |
| 7.5 | DLL management protocol (DLM)..... | 58 |
| 7.5.1 | Overview | 58 |
| 7.5.2 | Primitive definitions | 58 |
| 7.5.3 | DLM state machine (DLM_SM) | 59 |
| | Bibliography..... | 61 |
| | Figure 1 – Relationships of DLSAPs, DLSAP-addresses and group DL-addresses | 16 |
| | Figure 2 – Ring control in Type 25 network | 18 |
| | Figure 3 – Communication ranges of Type 25 frames | 19 |
| | Figure 4 – Priority control with VLAN of Type 25 network..... | 20 |
| | Figure 5 – The mechanism of transmission delay in a node | 20 |
| | Figure 6 – The maximum delay in Type 25 network..... | 21 |
| | Figure 7 – Data-Link layer internal architecture..... | 22 |
| | Figure 8 – Type 25 fieldbus DLPDU frame format | 25 |
| | Figure 9 – RCL frame format..... | 26 |
| | Figure 10 – State transition diagram of RHE_SM-A..... | 33 |
| | Figure 11 – State transition diagram of RHE_SM-B..... | 36 |
| | Figure 12 – The state diagram of RCLNode_SM | 39 |
| | Figure 13 – The state diagram of RCLTR_SM..... | 46 |
| | Figure 14 – The state diagram of RTTR_SM | 51 |
| | Figure 15 – The state diagram of TRC_SM | 52 |
| | Figure 16 – The state diagram of DLM_SM | 59 |
| | Table 1 – State transition descriptions | 16 |
| | Table 2 – Descriptions of state machine elements | 17 |
| | Table 3 – Conventions used in state machine | 17 |
| | Table 4 – Characteristics of the node states | 18 |
| | Table 5 – Characteristic of the frame classes..... | 19 |
| | Table 6 – VLAN priority mapping of Type 25 network..... | 19 |
| | Table 7 – Data-link layer components | 22 |
| | Table 8 – Destination address format..... | 25 |
| | Table 9 – VLAN tag format..... | 26 |
| | Table 10 – Types and classes of RCL frames | 27 |

| | |
|--|----|
| Table 11 – Structure of RCL header..... | 28 |
| Table 12 – Class field format | 28 |
| Table 13 – Destination address field format | 28 |
| Table 14 – Source address field format..... | 29 |
| Table 15 – CMD field format | 29 |
| Table 16 – The primitives and parameters for DLS-user interface | 30 |
| Table 17 – Parameters used with primitives exchanged between RCLC and DLS-user | 30 |
| Table 18 – The primitives and parameters for TRC interface..... | 31 |
| Table 19 – Parameters used with primitives exchanged between RCLC and TRC | 31 |
| Table 20 – The primitives and parameters for DLM interface..... | 32 |
| Table 21 – Parameters used with primitives exchanged between RCLC and DLM..... | 32 |
| Table 22 – Transitions of RHE_SM-A at RCL communication..... | 33 |
| Table 23 – Transitions of RHE_SM-B at RCL communication..... | 36 |
| Table 24 – Transitions of RCLNode_SM at RCL communication | 39 |
| Table 25 – Transitions of RCLTR_SM at RCL communication | 47 |
| Table 26 – RCLC function table | 48 |
| Table 27 – The primitives and parameters for DLS-user interface | 49 |
| Table 28 – Parameters used with primitives exchanged between RTC and DLS-user..... | 49 |
| Table 29 – The primitives and parameters for TRC interface..... | 49 |
| Table 30 – Parameters used with primitives exchanged between RTC and TRC | 50 |
| Table 31 – The primitives and parameters for DLM interface..... | 50 |
| Table 32 – Parameters used with primitives exchanged between RTC and DLM | 50 |
| Table 33 – Transitions of RTTR_SM at RT communication | 51 |
| Table 34 – RTC function table..... | 51 |
| Table 35 – The primitives and parameters for DLM interface..... | 52 |
| Table 36 – Parameters used with primitives exchanged between TRC and DLM | 52 |
| Table 37 – Transitions of TRC_SM | 53 |
| Table 38 – TRC function table..... | 57 |
| Table 39 – Primitives exchanged between DLM and DLS-user..... | 58 |
| Table 40 – Parameters used with primitives exchanged between DLM and DLS-user..... | 59 |
| Table 41 – Transitions of DLM_SM | 60 |

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
 FIELDBUS SPECIFICATIONS –**
**Part 4-25: Data-link layer protocol specification –
 Type 25 elements**
FOREWORD

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

International Standard IEC 61158-4-25 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

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

| | |
|--------------|------------------|
| FDIS | Report on voting |
| 65C/946/FDIS | 65C/955/RVD |

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

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

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

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.

INTRODUCTION

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

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

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

This document is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this document together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems 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 from 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 a patent concerning Type 25 elements and possibly other types given in this document as follows:

| | |
|------------------|--|
| JP4074631 [HI] | Transmission line system, frame transmitter therein, and transmission line switching method |
| JP4653800 [HI] | Transmission line system, frame transmission apparatus, method and program for switching transmission line in transmission line system |
| JP4944986 [HI] | Transmission line system and transmission line construction method |
| CN1964307 [HI] | Transfer path system and frame transfer device in same system, transfer path handover method and system |
| CN101515887 [HI] | Transmission line system, frame transmitter therein, transmission line switching method and program |

IEC takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right 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 this patent right is registered with IEC. Information may be obtained from:

[HI] Hitachi, Ltd.
Intellectual Property Division
20-2, Saiwai-cho 1-chome, Hitachi-shi,
Ibaraki-ken, 317-0073 Japan

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://patents.iec.ch>) 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 4-25: Data-link layer protocol specification – Type 25 elements

1 Scope

1.1 General

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

This protocol provides communication opportunities to all participating data-link entities

- a) in a synchronously-starting cyclic manner, according to a pre-established schedule, and
- b) in a cyclic or acyclic asynchronous manner, as requested each cycle by each of those data-link entities.

Thus this protocol can be characterized as one which provides cyclic and acyclic access asynchronously but with a synchronous restart of each cycle.

1.2 Specifications

This document specifies

- a) procedures for the timely transfer of data and control information from one data-link user entity to a peer user entity, and among the data-link entities forming the distributed datalink service provider;
- b) procedures for giving communications opportunities to all participating DL-entities, sequentially and in a cyclic manner for deterministic and synchronized transfer at cyclic intervals up to one millisecond;
- c) procedures for giving communication opportunities available for time-critical data transmission together with non-time-critical data transmission without prejudice to the time-critical data transmission;
- d) procedures for giving cyclic and acyclic communication opportunities for time-critical data transmission with prioritized access;
- e) procedures for giving communication opportunities based on ISO/IEC/IEEE 8802-3 medium access control, with provisions for nodes to be added or removed during normal operation;
- f) the structure of the fieldbus DLPDUs used for the transfer of data and control information by the protocol of this document, and their representation as physical interface data units.

1.3 Procedures

The procedures are defined in terms of

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

1.4 Applicability

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

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

1.5 Conformance

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

2 Normative references

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

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

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

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

ISO/IEC/IEEE 8802-3:2017, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Standard for Ethernet*

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

IEEE Std 802.1D, *IEEE Standard for Local and metropolitan area networks – Media access Control (MAC) Bridges*, available at <http://www.ieee.org> [viewed 2018-09-17]

IEEE Std 802.1Q, *IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks*, available at <http://www.ieee.org> [viewed 2018-09-17]

3 Terms, definitions, symbols, abbreviations and conventions

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

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

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

3.1 Reference model terms and definitions

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

| | |
|--|------------------|
| 3.1.1 DL-address | [ISO/IEC 7498-3] |
| 3.1.2 DL-address-mapping | [ISO/IEC 7498-1] |
| 3.1.3 called-DL-address | [ISO/IEC 7498-3] |
| 3.1.4 calling-DL-address | [ISO/IEC 7498-3] |
| 3.1.5 centralized multi-end-point-connection | [ISO/IEC 7498-1] |
| 3.1.6 DL-connection | [ISO/IEC 7498-1] |
| 3.1.7 DL-connection-end-point | [ISO/IEC 7498-1] |
| 3.1.8 DL-connection-end-point-identifier | [ISO/IEC 7498-1] |
| 3.1.9 DL-connection-mode transmission | [ISO/IEC 7498-1] |
| 3.1.10 DL-connectionless-mode transmission | [ISO/IEC 7498-1] |
| 3.1.11 correspondent (N)-entities correspondent DL-entities (N=2) correspondent Ph-entities (N=1) | [ISO/IEC 7498-1] |
| 3.1.12 DL-duplex-transmission | [ISO/IEC 7498-1] |
| 3.1.13 (N)-entity DL-entity (N=2) Ph-entity (N=1) | [ISO/IEC 7498-1] |
| 3.1.14 DL-facility | [ISO/IEC 7498-1] |
| 3.1.15 flow control | [ISO/IEC 7498-1] |
| 3.1.16 (N)-layer DL-layer (N=2) Ph-layer (N=1) | [ISO/IEC 7498-1] |
| 3.1.17 layer-management | [ISO/IEC 7498-1] |
| 3.1.18 DL-local-view | [ISO/IEC 7498-3] |
| 3.1.19 DL-name | [ISO/IEC 7498-3] |
| 3.1.20 naming-(addressing)-domain | [ISO/IEC 7498-3] |
| 3.1.21 peer-entities | [ISO/IEC 7498-1] |
| 3.1.22 primitive name | [ISO/IEC 7498-3] |
| 3.1.23 DL-protocol | [ISO/IEC 7498-1] |
| 3.1.24 DL-protocol-connection-identifier | [ISO/IEC 7498-1] |
| 3.1.25 DL-protocol-data-unit | [ISO/IEC 7498-1] |
| 3.1.26 DL-relay | [ISO/IEC 7498-1] |
| 3.1.27 reset | [ISO/IEC 7498-1] |
| 3.1.28 responding-DL-address | [ISO/IEC 7498-3] |
| 3.1.29 routing | [ISO/IEC 7498-1] |
| 3.1.30 segmenting | [ISO/IEC 7498-1] |
| 3.1.31 (N)-service DL-service (N=2) Ph-service (N=1) | [ISO/IEC 7498-1] |
| 3.1.32 (N)-service-access-point DL-service-access-point (N=2) Ph-service-access-point (N=1) | [ISO/IEC 7498-1] |

| | | |
|--------|---|------------------|
| 3.1.33 | DL-service-access-point-address | [ISO/IEC 7498-3] |
| 3.1.34 | DL-service-connection-identifier | [ISO/IEC 7498-1] |
| 3.1.35 | DL-service-data-unit | [ISO/IEC 7498-1] |
| 3.1.36 | DL-simplex-transmission | [ISO/IEC 7498-1] |
| 3.1.37 | DL-subsystem | [ISO/IEC 7498-1] |
| 3.1.38 | systems-management | [ISO/IEC 7498-1] |
| 3.1.39 | DLS-user-data | [ISO/IEC 7498-1] |

3.2 Service convention terms and definitions

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

| | |
|--------|--|
| 3.2.1 | acceptor |
| 3.2.2 | asymmetrical service |
| 3.2.3 | confirm (primitive); requestor.deliver (primitive) |
| 3.2.4 | deliver (primitive) |
| 3.2.5 | DL-confirmed-facility |
| 3.2.6 | DL-facility |
| 3.2.7 | DL-local-view |
| 3.2.8 | DL-mandatory-facility |
| 3.2.9 | DL-non-confirmed-facility |
| 3.2.10 | DL-provider-initiated-facility |
| 3.2.11 | DL-provider-optional-facility |
| 3.2.12 | DL-service-primitive; primitive |
| 3.2.13 | DL-service-provider |
| 3.2.14 | DL-service-user |
| 3.2.15 | DLS-user-optional-facility |
| 3.2.16 | indication (primitive); acceptor.deliver (primitive) |
| 3.2.17 | multi-peer |
| 3.2.18 | request (primitive); requestor.submit (primitive) |
| 3.2.19 | requestor |
| 3.2.20 | response (primitive); acceptor.submit (primitive) |
| 3.2.21 | submit (primitive) |
| 3.2.22 | symmetrical service |
| 3.3 | Terms and definitions |

3.3.1

blocking

port state which does not participate in frame communication

3.3.2

class

identifiers that designate communication range of the RCL frame and the other frames

3.3.3

control communication

non-real-time acyclic data communication for higher priority applications and node control communication

3.3.4

cyclic communication

periodic data communication for real-time communication

3.3.5

DLCEP-address

DL-address which designates either

- a) one peer DL-connection-end-point, or
- b) one multi-peer publisher DL-connection-end-point and implicitly the corresponding set of subscriber DL-connection-end-points where each DL-connection-end-point exists within a distinct DLSAP and is associated with a corresponding distinct DLSAP-address.

3.3.6

DLSAP

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

3.3.7

DL(SAP)-address

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

Note 1 to entry: This terminology is chosen because ISO/IEC 7498-3 does not permit the use of the term DLSAP-address to designate more than a single DLSAP at a single DLS-user.

3.3.8

(individual) DLSAP-address

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

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

3.3.9

extended link

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

3.3.10

frame

denigrated synonym for DLPDU

3.3.11

group DL-address

DL-address that potentially designates more than one DLSAP within the extended link

Note 1 to entry: A single DL-entity may have multiple group DL-addresses associated with a single DLSAP. A single DL-entity also may have a single group DL-address associated with more than one DLSAP.

3.3.12

information communication

non-real-time acyclic data communication for low priority applications

3.3.13**logical link down**

link status at which the port is in a blocking state and does not communicate all kinds of frames except RCL frames

3.3.14**logical link up**

link status at which the port communicates all kinds of frames

3.3.15**node**

single DL-entity as it appears on one local link

3.3.16**physical link down**

link status at which the port does not communicate the frames due to link down status defined in ISO/IEC/IEEE 8802-3

3.3.17**receiving DLS-user**

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

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

3.3.18**ring control (RCL) communication**

control communication of Type 25 DLL ring network using RCL frames and non-real-time

3.3.19**sending DLS-user**

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

3.3.20**station**

synonym for node

3.3.21**station address**

identifier address that designates the node of Type 25 network

3.4 Symbols and abbreviations

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

| | |
|-------|---|
| DL- | Data-link layer (as a prefix) |
| DLC | DL-connection |
| DLCEP | DL-connection-end-point |
| DLE | DL-entity (the local active instance of the data-link layer) |
| DLL | DL-layer |
| DLM | DL-management |
| DLME | DL-management Entity (the local active instance of DL-management) |
| DLMS | DL-management service |
| DLPCI | DL-protocol-control-information |
| DLPDU | DL-protocol-data-unit |
| DLS | DL-service |

| | |
|-------|---|
| DLSAP | DL-service-access-point |
| DLSDU | DL-service-data-unit |
| EGA | Edge-A node |
| EGB | Edge-B node |
| FIFO | First-in first-out (queuing method) |
| ISL | Isolate node |
| ITM | Intermediate node |
| LCA | Loop condition alert (Type 25 frame type) |
| LCC | Loop condition check (Type 25 frame type) |
| LCN | Loop condition notify (Type 25 frame type) |
| LLD | Logical link down |
| LLU | Logical link up |
| LNA | Loop notify answer (Type 25 frame type) |
| NNB | No neighborhood state |
| OSI | Open systems interconnection |
| Ph- | Physical layer (as a prefix) |
| PhE | Ph-entity (the local active instance of the Physical layer) |
| PhL | Ph-layer |
| PhS | Ph-service |
| PLD | Physical link down |
| PLU | Port A (B) link up state |
| QoS | Quality-of-service |
| RCL | Ring control |
| RCLC | RCL communication control |
| RHE | Rapid hello (Type 25 frame type) |
| RT | Real time |
| RTC | RT communication control |
| SCR | Station condition report (Type 25 frame type) |
| TRC | Transmit/Receive control |
| WLU | Wait link up state |

3.5 Common conventions

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

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

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

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

- the request primitive's input parameters;
- the request primitive's output parameters;
- the indication primitive's output parameters;
- the response primitive's input parameters;
- the confirm primitive's output parameters.

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

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

- M** – parameter is mandatory for the primitive;
- U** – parameter is a user option and may or may not be provided depending on the dynamic usage of the DLS-user. When not provided, a default value for the parameter is assumed;
- C** – parameter is conditional upon other parameters or upon the environment of the DLS user;
- (blank) – parameter is never present.

Some entries are further qualified by items in brackets. These may be

a) a parameter-specific constraint

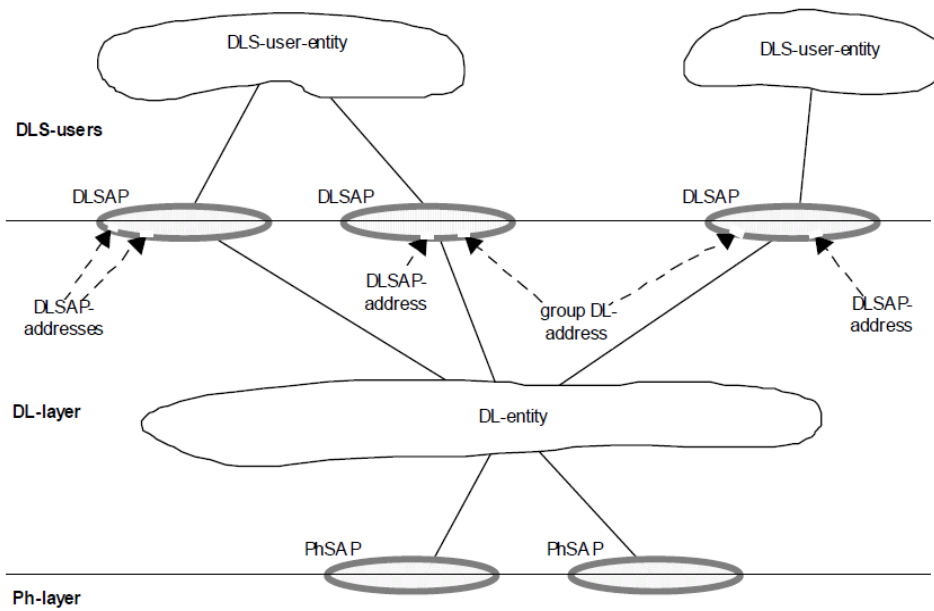
(=) indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table;

b) an indication that some note applies to the entry

(n) indicates that the following note n contains additional information pertaining to the parameter and its use.

In any particular interface, not all parameters need be explicitly stated. Some may be implicitly associated with the DLSAP at which the primitive is issued.

Figure 1 shows the relationships of DLSAPs, DLSAP-addresses and group DL-addresses.



- NOTE 1 DLSAPs and PhSAPs are depicted as ovals spanning the boundary between two adjacent layers.
- NOTE 2 DL-addresses are depicted as designating small gaps (points of access) in the DLL portion of a DLSAP.
- NOTE 3 A single DL-entity can have multiple DLSAP-addresses and group DL-addresses associated with a single DLSAP.

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

In the diagrams which illustrate these interfaces, dashed lines indicate cause-and-effect or time-sequence relationships, and wavy lines indicate that events are roughly contemporaneous.

3.6 Additional Type 25 conventions

3.6.1 Primitive conventions

The following notation, a shortened form of the primitive classes defined in 3.2, is used in the figures.

- req** request primitive
- ind** indication primitive
- cnf** confirm primitive (confirmation)
- rsp** response primitive

3.6.2 State machine conventions

The protocol sequences are described by means of state machines.

In state diagrams, states are represented as boxes and state transitions are shown as arrows.

Names of states and transitions of the state diagram correspond to the names in the state table. The textual listing of the state transitions is structured as shown in Table 1.

Table 1 – State transition descriptions

| # | Current state | Event /condition => actions | Next state |
|---|---------------|-----------------------------|------------|
|---|---------------|-----------------------------|------------|

Table 2 – Descriptions of state machine elements

| Description element | Meaning |
|-------------------------------|---|
| # | Number of the transition. |
| Current state, Next state | Names of the originating state and the target state of transition. |
| Event | Name or description of the trigger event that fire the transition. |
| /conditions | Boolean expression, which shall be true for the transition to be fired. |
| =>actions | List of assignments and service or function invocations. The action should be atomic. The preceding "=>" is not part of the action. |
| "/ conditions" may be omitted | |

The conventions used in the state machines are shown in Table 3.

Table 3 – Conventions used in state machine

| Convention | Meaning |
|------------|---|
| = | Substitution of the right side for the left side |
| == | A logical condition to indicate an item on the left is equal to an item on the right. |
| != | A logical condition to indicate an item on the left is not equal to an item on the right |
| < | A logical condition to indicate an item on the left is less than the item on the right |
| > | A logical condition to indicate an item on the left is greater than the item on the right |
| && | Logical "AND" |
| | Logical "OR" |
| ! | Negation operator |
| + - * / | Arithmetic operator |
| ++ | Increment operator |
| -- | Decrement operator |
| ; | Breakpoint |

4 Overview of the DL-protocol

4.1 General

A Type 25 fieldbus is based on the ISO/IEC/EEE 8802-3 but extended towards real-time Ethernet. Type 25 network controls the data traffic and assigns frame priorities with IEEE 802.1Q VLAN.

NOTE Any standard Ethernet silicon, infrastructure component or test and measurement equipment such as a network analyzer is applicable to Type 25 network.

4.2 Overview of the medium access control

4.2.1 General

Type 25 network provides mechanisms to achieve the following.

- a) It assigns priorities to four types of frames (RCL/Cyclic/Control/Information).
- b) The frames in Type 25 network are guaranteed real-time of cyclic communication to control the data traffic.

- c) It uses ISO/IEC/IEEE 8802-3 communication without needs for special functions for RTE communication such as time sharing or node synchronization.
- d) The ring topology is only permitted in Type 25 network. In order to control the network, it categorizes the frames into three classes according to their communication ranges.

4.2.2 Network topology

The ring topology shall be used in Type 25 network. The network shall be controlled and reconfigured by RCL frames. Type 25 ring network does not need concept of master/slave model because each node in Type 25 network reconfigures autonomously. To prevent frame loop, one section in the ring network is in the blocking state (see IEEE 802.1D) and the blocked section terminates all kinds of frames except RCL frame (as illustrated in Table 2).

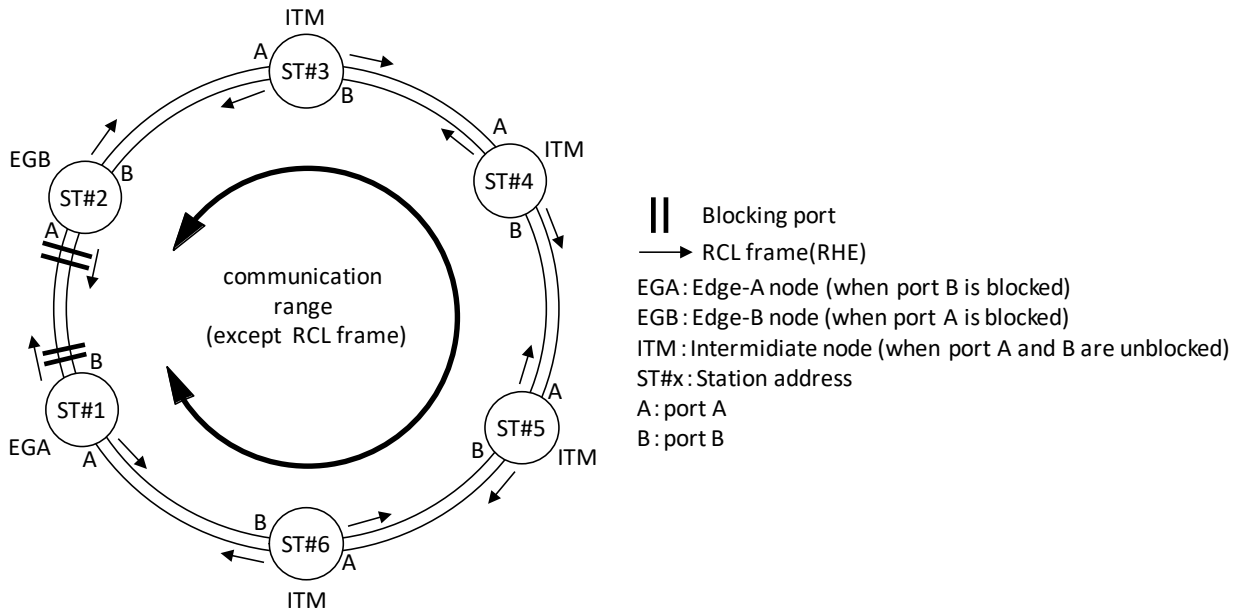


Figure 2 – Ring control in Type 25 network

Each node in the ring network has two ports A and B to connect to the neighboring nodes. Each node has its own station address. In Type 25 network, each node has four different node states depending on the blocking status of their ports. Table 4 shows the characteristics of the node states.

Table 4 – Characteristics of the node states

| Node states | Abbreviation | Description |
|---|--------------|--|
| Isolated node | ISL | Both port A and B are blocked in this state (e.g. not connect cables to other nodes) and the node is isolated from the Type 25 network. |
| Edge-A node (when port B is blocked) | EGA | Port B is blocked, port A is unblocked. The node works with Edge-A. There is only one node in a Type 25 network. |
| Edge-B node (when port A is blocked) | EGB | Port A is blocked, port B is unblocked. The node works with Edge-B. There is only one node in a Type 25 network. |
| Intermediate node (when port A, B are unblocked) | ITM | Both port A and B are unblocked. The node works with Edge-A and Edge-B. In a Type 25 network, there are an Edge-A node and an Edge-B node. The other nodes are this state. |

RCL frames are based on general ISO/IEC/IEEE 8802-3 frames. These frames have different ranges of communication which are given by “class”. Table 5 shows the characteristics of the frame class and Figure 3 shows the communication ranges in Type 25 network.

Table 5 – Characteristic of the frame classes

| Frame class | Description |
|-------------|---|
| Class 1 | This frame is used to communicate with the neighboring node only. It passes through blocking ports. RHE frame are RCL frames in this class. |
| Class 2 | This frame is used to communicate around the network. This frame is able to pass through blocking ports and the Edge-A node terminates the frames. LCC, LCA, LCN, LNA, and SCR frames are RCL frames in this class. |
| Class 3 | This frame is general Ethernet frame and terminated by blocking ports. The RT frames (Cyclic, Control, Information frames) are in this class. |

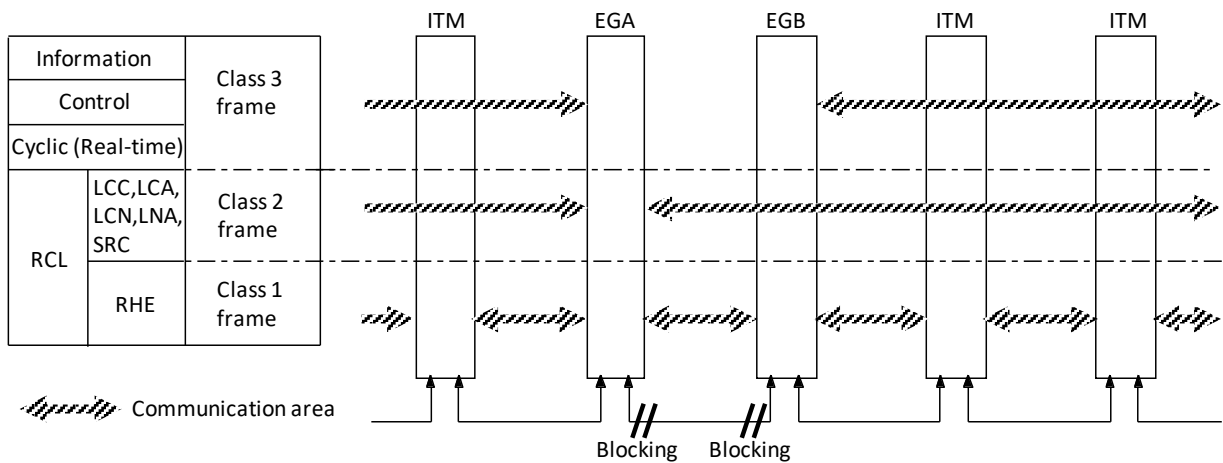


Figure 3 – Communication ranges of Type 25 frames

4.2.3 Priority control with VLAN

Table 6 shows the VLAN priority mapping of the Type 25 network. Type 25 network maps VLAN priority to four types communication, Ring control, Cyclic, Control, and Information communication. Ring control communication, Type 25 ring network reconfiguration frame is the highest priority. Cyclic communication, real-time communication frame is the second highest priority. Control communication is the third priority and Information is the lowest priority.

Table 6 – VLAN priority mapping of Type 25 network

| VLAN | Usage | VLAN priority |
|--------|----------------------------|---------------|
| VLAN-1 | Ring control communication | 7 (Highest) |
| VLAN-2 | Cyclic communication | 5 (High) |
| VLAN-3 | Control communication | 3 (Low) |
| VLAN-4 | Information communication | 1 (Lowest) |

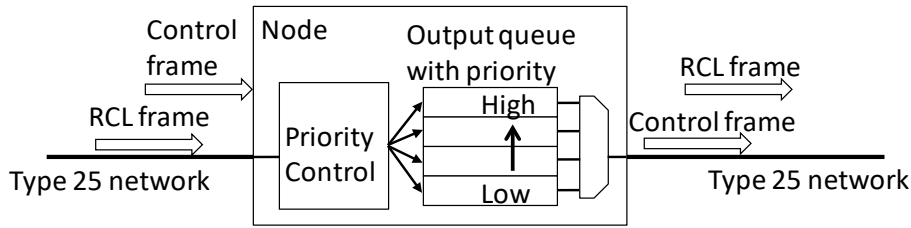


Figure 4 – Priority control with VLAN of Type 25 network

Each node has output queues with priority. The receiving frames are put into the output queues based on their frame priorities. The frame with higher priority will be forwarded neighboring node. In this mechanism, Type 25 network guarantees real-time cyclic communication.

4.2.4 The maximum delivery delay in Type 25 network

Type 25 network limits data traffic in the network to ensure real time communication. This maintains the communication delay of frames at an acceptable level. The traffic restriction of each node is determined by the total number of nodes in the network and cycle time using the node.

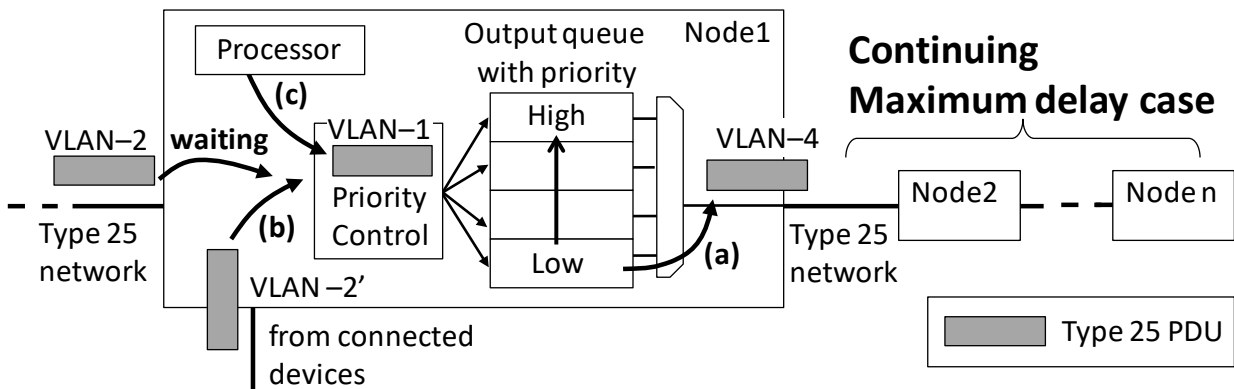


Figure 5 – The mechanism of transmission delay in a node

There are three causes of the delay regarding VLAN-2 (cyclic communication) frame in Type 25 network.

- a) VLAN-2 frame arrives at the time when the node is sending another frame. The VLAN-2 frame is forced to wait until the transmission is finished. This delay is caused by any priority frames.
- b) Before receiving a VLAN-2 frame, the node receives another VLAN-2 frame (VLAN-2' in Figure 5) from other ports. In this case, at the priority queue, the VLAN2 frame is put after the VLAN-2' frame.
- c) When the VLAN-2 frame is waiting for transmission due to the causes a) and b), the node receives higher priority frames (VLAN-1 frame). Since the node will transmit the highest priority frames first, the VLAN-2 frame is forced to wait until the transmission of higher priority frames is finished.

When three causes happen all at once in a node, the maximum delay occurs. The delivery time could be calculated in the case of the maximum delay occurring at all nodes in a Type 25 network.

4.2.5 Traffic control for real-time communication

Delivery time in Type 25 network is calculated based on the maximum delay time per node, the number of nodes, and the length of ring network.

The maximum delay of real-time communication (cyclic communication) between two nodes is calculated by taking into account to the four delay causes shown in Figure 6.

1) Stacking delay in sending node:

The delay caused by the transmission process. It depends on the implementation of the hardware and software.

2) Cable segment delay:

The delay caused by transmission cable constructing ring network. It is proportional to the cable length.

3) Transmission delay in transit node

The delay caused by transmission delay per transit node. It is proportional to the number of nodes in the transmission path. The detail is shown in 4.2.4.

4) Stacking delay in receiving node:

The delay caused by the receiving process. It depends on the implementation of the hardware and software.

The delivery time is calculated by the following equation.

$$(\text{delivery time}) = (1) + (2) \times \text{cable length} + (3) \times (\text{the number of nodes} - 2) + (4)$$

The cycle time of cyclic communication shall be set to larger than the delivery time to guarantee real-time communication of cyclic communication frames in Type 25 network.

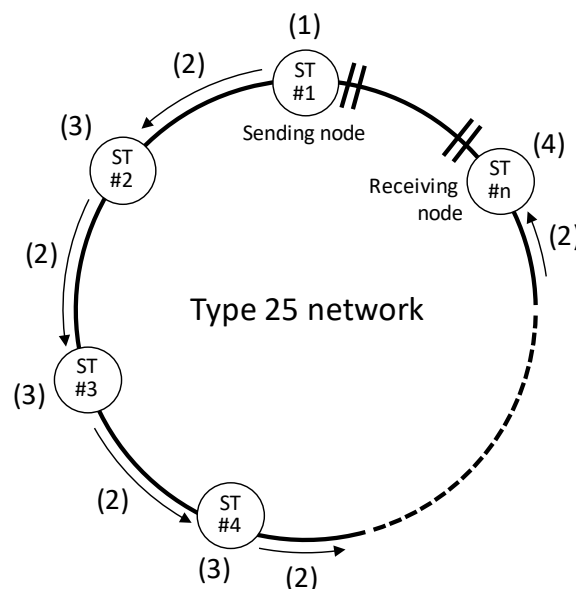


Figure 6 – The maximum delay in Type 25 network

4.3 Service assumed from PhL

Subclause 4.3 describes the assumed physical service (PhS) and the constraints used by the DLE. The Physical Service is assumed to provide the following service primitives specified by ISO/IEC/IEEE 8802-3:2017, Clause 6.

The assumed primitives of PhS are

- PhS-A_DATA.req;
- PhS-A_DATA.ind;
- PhS-B_DATA.req;
- PhS-B_DATA.ind.

4.4 DL Layer architecture

The Type 25 fieldbus DLL is modeled as a combination of control components of RCL communication control (RCLC), Real-time communication control (RTC), Transmit/Receive control (TRC) and DL-management (DLM).

The data-link layer is comprised of the components shown in Table 7.

Table 7 – Data-link layer components

| Components | Description |
|---------------------------------------|---|
| RCL communication control (RCLC) | At RCLC, DLSDU for the ring control (RCL) data is stored and transferred between the DLS-user and the TRC. |
| Real-time communication control (RTC) | At RTC, control data/information data is stored and transferred between DLS-user and the TRC. |
| Transmit/Receive control (TRC) | At TRC, DLSDU is stored and transferred from the RCLC/RTC to Ph-Layer. DLPDU is stored and transferred from Ph-Layer to the RCLC/RTC. |
| DL-management (DLM) | The DL-management 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, is shown in Figure 7. The arrowheads illustrate the primary direction of the flow of data and control.

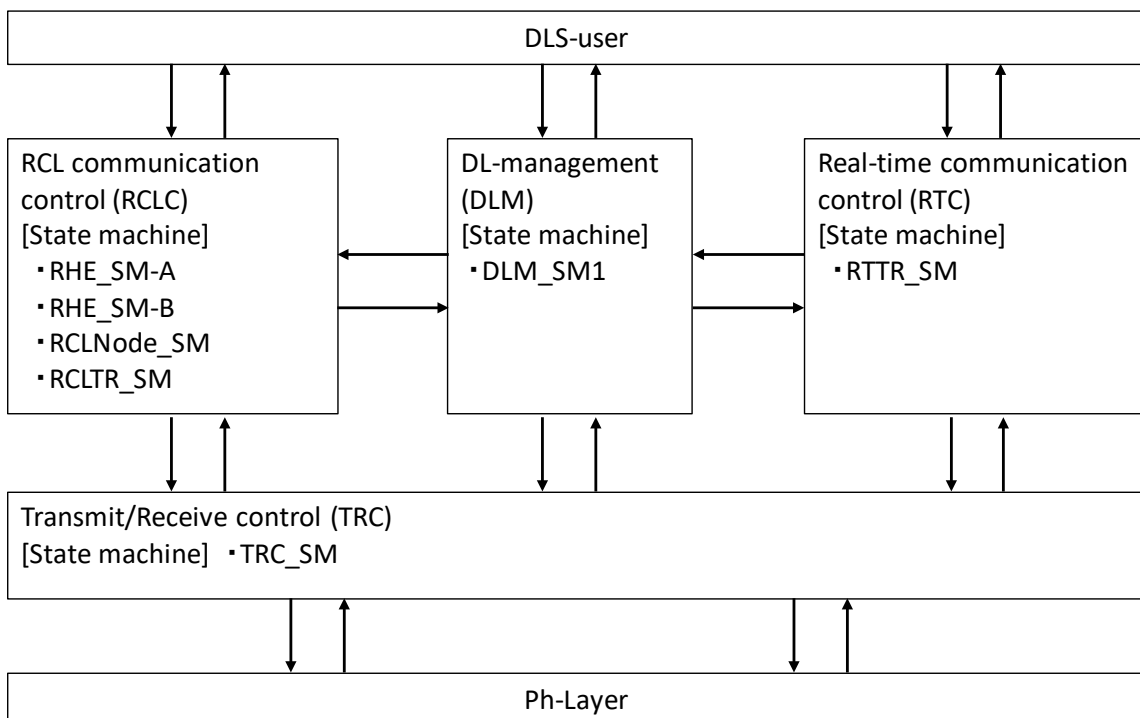


Figure 7 – Data-Link layer internal architecture

4.5 Local parameters and variables

4.5.1 Overview

This specification uses DLS-user request parameters P(...) and local variables V(...) as a means of clarifying the effect of certain actions and the conditions under which those actions are valid, local timers T(...) as a means of monitoring actions of the distributed DLS-provider and of ensuring a local DLE response to the absence of those actions, and local counters C(...) for performing rate measurement functions.

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;

DL-management may change the values of configuration variables.

4.5.2 Variables, parameter, counter and timer

4.5.2.1 P(RHE_RxCK_NUM)

This parameter indicates the expiration time of not receiving RHE frames.

4.5.2.2 P(RHE_LKUP_NUM)

This parameter indicates the number of receiving RHE frames for logical linkup.

4.5.2.3 P(LCC_STOP_NUM)

This parameter indicates the threshold number to stop the LCC frame transmission.

4.5.2.4 C(PA_RHE_LKUP), C(PB_RHE_LKUP)

These counters indicate the count of continuously receiving RHE frames at port A and port B, respectively.

4.5.2.5 C(PA_RHE_RxCK), C(PB_RHE_RxCK)

These counters are used to check whether continuously received or dropped RHE frames. Each value is decreased in T(PA_RHE_TIME) or T(PB_RHE_TIME).

4.5.2.6 C(PA_Rx-RHE_NotMatch), C(PB_Rx-RHE_NotMatch)

These counters are used to detect the change from neighborhood node to another node.

4.5.2.7 C(Rx_Counter)

This counter indicates the number of receiving LCC frames by own node.

4.5.2.8 T(PA_RHE_TIME), T(PB_RHE_TIME)

T(PA_RHE_TIME) and T(PB_RHE_TIME) are used for fixed cycle to do RHE operation.

4.5.2.9 V(PA_Blocking), V(PB_Blocking)

These variables hold and designate the blocking status at the ports respectively. When the port is blocked, these variables are set to "True", otherwise "False".

4.5.2.10 V(LCN_Pri)

This variable is a unique value used to decide an Edge-A node in the network. This variable is used when the node sends LCN frames.

4.5.2.11 V(LNA_Pri)

This variable is a unique value used to decide an Edge-A node in the network. This variable is used when the node sends LNA frames.

4.5.2.12 V(PA_NBST_LKST), V(PB_NBST_LKST)

These variables hold and designate the link status of a neighborhood node. It takes three types value (NNB, WLU, and PLU). These values are updated by receiving the RHE frames.

4.5.2.13 V(PA_NBST_NDST), V(PB_NBST_NDST)

This variable holds and designates the node status of a neighborhood node. It takes four types value (ISL, EGA, EGB, and ITM) and updates the value by receiving the RHE frames.

4.5.2.14 V(PA_NBST_RCLADD), V(PB_NBST_RCLADD)

These variables hold and designate the address of a neighborhood node. These values are updated by receiving the RHE frames.

4.5.2.15 V(PA_RHE_CYCLE), V(PB_RHE_CYCLE)

These variables hold and designate the cycle time to do RHE operation.

4.5.2.16 V(PA_RHE_RxSeq), V(PB_RHE_RxSeq)

These variables hold and designate the sequence number of receiving RHE frames. If RHE frames from neighborhood node are received continuously, these values increase.

4.5.2.17 V(PA_STATUS), V(PB_STATUS)

These variables hold and designate the port link status of own node. They take three values (NNB, WLU, and PLU).

5 General structure and encoding of PhPDUs and DLPDU and related elements of procedure

5.1 Overview

The DLL and its procedures are necessary to provide the services offered to the DLS user by using the services available from the PhL. Clause 5 describes the structure and semantics of DLPDU and the procedure, commonly used in this specification.

5.2 Common MAC frame structure, encoding and elements of procedure

5.2.1 MAC frame structure

Figure 8 shows the Type 25 fieldbus DLPDU.

| | | | | | | | | |
|-----------------|------------|----------------------------|-----------------------|-----------------|---------------------|---------------|---------|------------|
| Preamble (7) | SFD (1) | Destination address (6) | Source address (6) | VLAN tag (4) | Length/ Type (2) | DLS-user data | Padding | FCS (4) |
|-----------------|------------|----------------------------|-----------------------|-----------------|---------------------|---------------|---------|------------|

(*the number in parenthesis indicates length of the field in octets)

Figure 8 – Type 25 fieldbus DLPDU frame format

5.2.2 Elements of the MAC frame

5.2.2.1 Preamble

The preamble field is identical to ISO/IEC/IEEE 8802-3:2017, Clause 3. The preamble field is a 56-bit field that is used to allow the physical signaling part circuitry to reach its steady state synchronization with the receiving frame timing.

The preamble pattern is:

“10101010 10101010 10101010 10101010 10101010 10101010 10101010.”

5.2.2.2 Start frame delimiter (SFD)

The Start Frame Delimiter (SFD) is identical to ISO/IEC/IEEE 8802-3:2017, Clause 3. The SFD field is the sequence of bit pattern “10101011”. It immediately follows the preamble pattern and indicates the start of a frame.

5.2.2.3 Destination address

The Destination address is identical to ISO/IEC/IEEE 8802-3:2017, Clause 3. This address field is a 48-bit in length. The frames received and sent at each node are classified into three classes (class 1, 2, and 3) based on the communication range. The detail of the class is shown in 4.2.2.

The frame class is distinguished by destination address. Table 8 shows the destination address of three types of classes.

Table 8 – Destination address format

| Frame class | Frame type | Destination address |
|-------------|----------------------------|--------------------------------------|
| Class 1 | RCL (RHE) | 01-80-C2-00-00-0F |
| Class 2 | RCL (LCC/LCA/LCN/LNA/SCR) | 01-80-C2-00-00-0E |
| Class 3 | Cyclic/Control/Information | Destination MAC address of the DLSDU |

5.2.2.4 Source address

Source address shall contain the node address of the source DLE.

5.2.2.5 VLAN tag

VLAN tag uses IEEE 802.1Q VLAN tag, and indicates frame type (RCL, cyclic, control, and information) designated VID and frame priority designated PCP. Table 9 shows the VLAN tag format.

Table 9 – VLAN tag format

| Frame type | TPID (2 octets) | TCI (2 octets) | | |
|-------------|--------------------|-------------------|----------------|-----------------|
| | | PCP (3 bit) | CFI (1 bit) | VID (12 bit) |
| RCL | 0x8100 | 111 | 0 | 0xFFB |
| Cyclic | 0x8100 | 101 | 0 | 0xFFC |
| Control | 0x8100 | 011 | 0 | 0xFFD |
| Information | 0x8100 | 001 | 0 | 0xFFE(*) |

(*): Information frames may use other VID than 0xFFE(4094) from 1 to 50 and designate multiple VIDs.

5.2.2.6 Length/Type

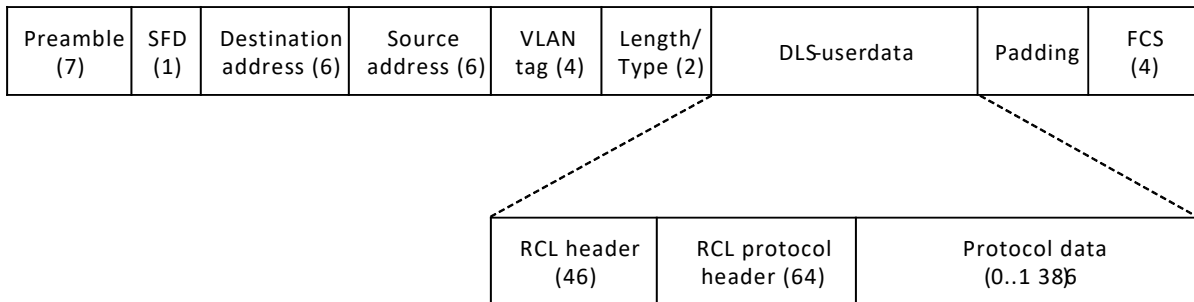
The Length/type is different for frame types.

- RCL frame is set to the frame length. It is different for the RCL_type.
- Cyclic frame is set to 0x0800 (designated IP frame).
- Control and information frames are set to protocol ID that corresponds to PDU data.

5.2.2.7 DLS-user data field

5.2.2.7.1 RCL frame

Figure 9 shows the RCL frame format. RCL frame has three segments, RCL header, RCL protocol header, and Protocol data in DLS-user data. Each segment contains information which is used to control the ring network. The information differs according to RCL frame types.



Key: (#) the number in parenthesis indicates length of the field in octets

Figure 9 – RCL frame format

5.2.2.7.2 RT frame

The RT frames are identical to the ISO/IEC/IEEE 8802-3.

5.2.2.8 Padding

Padding is inserted into the frame so that the data length is always a multiple of 32 bits.

5.2.2.9 Field check sequence (FCS)

The frame check sequence (FCS) construction, polynomial and expected residual are identical to ISO/IEC/IEEE 8802-3:2017, Clause 3.

6 DLPDU-specific structure, encoding and elements of procedure

6.1 General

Clause 6 defines the structure, contents and encoding of each type and format of the RCL communication DLPDU, and specifies elements of procedure for the DLPDU. The RT communication DLPDU is not described in Clause 6 since its frame is identical to the ISO/IEC/IEEE 8802-3.

6.2 Structure of the RCL DLPDU

Frame of class 1 is used to communicate to the neighboring node. Frame of class 2 is used to communicate between nodes around the network. For the RCL frame, there are one kind of class 1 frame and five kinds of class 2 frame. Table 10 shows the types and classes of RCL frames.

Table 10 – Types and classes of RCL frames

| Frame class | Frame name | Description |
|---|------------------------------------|---|
| Class 1 | RHE (Rapid Hello) | The RHE frame confirms the neighboring node and transmission path. |
| Class 2 | LCC (*) (Loop Condition Check) | The LCC frame is used to determine autonomously the single Edge-A node, which exists in the normal ring network. |
| Class 2 | LCA (*) (Loop Condition Alert) | If a node's port B is physical or logical link down, the node's LCA frame declares that it has higher priority to be Edge-A node. |
| Class 2 | LCN (*) (Loop Condition Notify) | A node's LCN frame declares that its own port B changes to physical link down state. |
| Class 2 | LNA (*) (Loop Notify Answer) | The LNA frame is sent to the node which sent LNA frame. Only Edge-B node sends this frame. |
| Class 2 | SCR (Station Condition Report) | The SCR frame notifies the event of change in network topology to all nodes in the ring network. |
| (*)LCC, LCA, LCN, and LNA have different priority to select an Edge-A node: LCN > LCA > LCC (high priority) > LCC (low priority) | | |

6.2.1 RCL header

6.2.1.1 General

The structure of RCL DLPDU is shown in Table 11.

Table 11 – Structure of RCL header

| RCL header field | | Size (Octet) |
|---------------------|-----------------|--------------|
| Class | | 2 |
| Destination address | Priority | 1 |
| | Station address | 1 |
| | MAC address | 6 |
| Source address | Priority | 1 |
| | Station address | 1 |
| | MAC address | 6 |
| CMD | | 4 |
| Sequence number | | 4 |
| Reserved | | 20 |

6.2.1.2 Class

There are two kinds of class in RCL frame as shown in Table 12.

- Class 1: used to communicate to the neighboring node;
- Class 2: used to communicate between nodes around the network.

Table 12 – Class field format

| Frame class | Value(2 octets) |
|-------------|-----------------|
| Class 1 | 0x0001 |
| Class 2 | 0x0002 |

6.2.1.3 Destination address

Table 13 shows the destination address field format. Set value of priority, station address, MAC address, depending on RCL_type.

Table 13 – Destination address field format

| Item | RCL_type | |
|-----------------|-------------------|--|
| | RHE/LCC/LCN/SCR | LCA/LNA |
| Priority | 0x00 | Set to the source address that receives RCL frame (LCA: received LCC frame, LNA: received LCN frame) |
| Station address | 0xFF | |
| MAC address | FF-FF-FF-FF-FF-FF | |

6.2.1.4 Source address

All RCL frames are set to the data shown in Table 14.

Table 14 – Source address field format

| Item | Value |
|-----------------|-----------------------------|
| Priority | 0x00 |
| Station address | Station address of the node |
| MAC address | MAC address of the node |

6.2.1.5 CMD

The CMD is used to identify the RCL frame. This area is different according to the frame type (RCL_type) as shown in Table 15.

Table 15 – CMD field format

| Bit | Item | Value | | | | | |
|----------|-----------------|-------|-----|-----|-----|-----|-----|
| | | RHE | LCC | LCA | LCN | LNA | SCR |
| [3..0] | Reserved | 1 | | | | | |
| [7..4] | Unique number 1 | 0 | 0 | 1 | 2 | 3 | 0 |
| [11..8] | Unique number 2 | 0 | 1 | 1 | 1 | 1 | 3 |
| [15..12] | Reserved | 0 | | | | | |
| [23..16] | Frame class | 1 | 2 | 2 | 2 | 2 | 2 |
| [31..24] | Reserved | 0 | | | | | |

6.2.1.6 Sequence number

Sequence number is used to detect loss and duplication of received frames. Its value ranges from 0x00000000 to 0xFFFFFFFF with an increment of 1.

7 DLE elements of procedure**7.1 Overview**

In Subclause 7.2, the operation of RCLC, RTC, TRC, and DLM are described.

7.2 RCL communication control (RCLC)**7.2.1 General**

RCLC provides the following functions.

- Control link status of communication with neighboring node using RHE frame;
- Control node status using class 2 RCL frames;
- Send and receive RCL frames.

7.2.2 Primitive definitions**7.2.2.1 Primitive definitions between RCLC and DLS-user**

Table 16 summarizes all primitives exchanged between the RCLC and the DLS-user.

Table 16 – The primitives and parameters for DLS-user interface

| Primitive name | Source | Associated parameters | Description |
|----------------|----------|--|--|
| DL-RCL.req | DLS-user | (in S_add, PortNum, Frame_pri, RCL_type, DLSDU) | Transmits a RCL frame from the DLS-user. The RCL frame is carried to TRC with the class parameter. |
| DL-RCL.cnf | RCLC | (out Status) | Responds to complete DLSDU transmission and reports the status to DLS-user. |
| DL-RCL.ind | RCLC | (out S_add, PortNum, Frame_pri, RCL_type, DLSDU) | Carries forward a received RCL frame to the DLS-user. |

The parameters used with the primitives exchange between the RCLC and the DLS-user are described in Table 17.

Table 17 – Parameters used with primitives exchanged between RCLC and DLS-user

| Parameter name | Description |
|----------------|--|
| S_add | The S_add parameter specifies the DL-address of the publisher. |
| PortNum | This parameter specifies the transmit/receive port of the publisher. |
| Frame_pri | This parameter specifies the priority with VLAN of the ring network. |
| RCL_type | This parameter specifies the RCL type of the RCL frame. |
| DLSDU | This parameter specifies the information of the RCL frame. |
| Status | This parameter allows DLMS-user to determine whether the requested DLS was provided successfully, or failed due to a particular reason. The value conveyed in this parameter is as follows: "OK – successfully completed"; "Failure – terminated before completion". |

7.2.2.2 Primitive definitions between RCLC and TRC

Table 18 summarizes all primitives exchanged between the RCLC and the TRC.

Table 18 – The primitives and parameters for TRC interface

| Primitive name | Source | Associated parameters | Description |
|----------------|--------|--|---|
| RCLC-Tx.req | RCLC | D_add, S_add, Tx_port, Frame_pri, RCL_type, DLSDU | Transmits a RCL frame to TRC with associated parameters. The RCLC attaches the Tx_port parameter (port A, B or both). |
| RCLC-Rx.ind | TRC | D_add, S_add, Rx_port, Frame_pri, RCL_type, DLSDU | Carries forward a received RCL frame from the TRC. |
| RCLC-SetPT.req | RCLC | BLKport, Blocking | Requests to change the blocking status of a port to the TRC. |

The parameters used with the primitives exchange between the RCLC and the TRC are described in Table 19.

Table 19 – Parameters used with primitives exchanged between RCLC and TRC

| Parameter name | Description |
|----------------|--|
| D_add | The D_add parameter specifies the DL-address of the subscriber. |
| S_add | The S_add parameter specifies the DL-address of the publisher. |
| Tx_port | This parameter specifies the transmit port of the publisher. |
| Rx_port | This parameter specifies the receive port of the publisher. |
| Frame_pri | This parameter specifies the priority with VLAN of the ring network. |
| RCL_type | This parameter specifies the RCL type of the RCL frame. |
| DLSDU | This parameter specifies the information of the RCL frame. |
| BLKport | This parameter specifies the blocking port in itself. |
| Blocking | This parameter specifies the blocking status in itself. |

7.2.2.3 Primitive definitions between RCLC and DLM

Table 20 summarizes all primitives exchanged between the RCLC and the DLM.

Table 20 – The primitives and parameters for DLM interface

| Primitive name | Source | Associated parameters | Description |
|----------------|--------|--|---|
| RCL_STOP.ind | RCLC | RCL_Indtype | Indicates a stop of the RCL frame to DLM based on the ring control state machine. |
| RCL_START.ind | RCLC | RCL_Indtype, RCL_IndDA, RCL_IndPri, RCL_IndPort | Indicates a start of the RCL frame to DLM based on the ring control state machine. |
| Node_ST.ind | RCLC | Node_ST, PortA_ST, PortB_ST | Indicates a change of the node or port status to DLM based on the ring control state machine. |

The parameters used with the primitives exchange between the RCLC and the DLM are described in Table 21.

Table 21 – Parameters used with primitives exchanged between RCLC and DLM

| Parameter name | Description |
|----------------|---|
| RCL_Indtype | This parameter specifies the RCL type of the requested RCL frames for the DLM due to the change of the node status. |
| RCL_IndDA | This parameter specifies the DL-address of the requested RCL frames for the DLM due to the change of the node status. |
| RCL_IndPri | This parameter specifies the priority with VLAN of the requested RCL frames for the DLM due to the change of the node status. |
| RCL_IndPort | This parameter specifies the transmission port of the requested RCL frames for the DLM due to the change of the node status. |
| Node_ST | This parameter contains the node status (ISL/EGA/EGB/ITM) for the DLM due to the change of the node status. |
| PortA_ST | This parameter contains the status of port A on own node for the DLM due to the change of the node status. |
| PortB_ST | This parameter contains the status of port B on own node for the DLM due to the change of the node status. |

7.2.3 RCLC state machine

7.2.3.1 RHE state machine of port A and B (RHE_SM-A/RHE_SM-B)

The RHE state machine is used to confirm the link status of communication with the neighboring node. There are two state machines on a node, one for port A (RHE_SM-A) and one for port B (RHE_SM-B). The state is updated when RHE is received and the link state between the node and the neighboring node is confirmed. These state machines have the following states.

- No_Neighborhood: The node does not connect to the neighboring node nor does it receive the RHE frame from the neighboring node.
- Wait_Linkup: The node waits to link up with the neighboring node. Although this state is “link up” status in ISO/IEC/IEEE 8802-3, the node should not communicate by any frames except the RHE frame to the neighboring node.
- Port_Linkup: The node can communicate using any ISO/IEC/IEEE 8802-3 frame to the neighboring node.

7.2.3.1.1 State machine RHE_SM-A

Figure 10 shows the state diagram of RHE_SM-A, and Table 22 is the state table of port A.

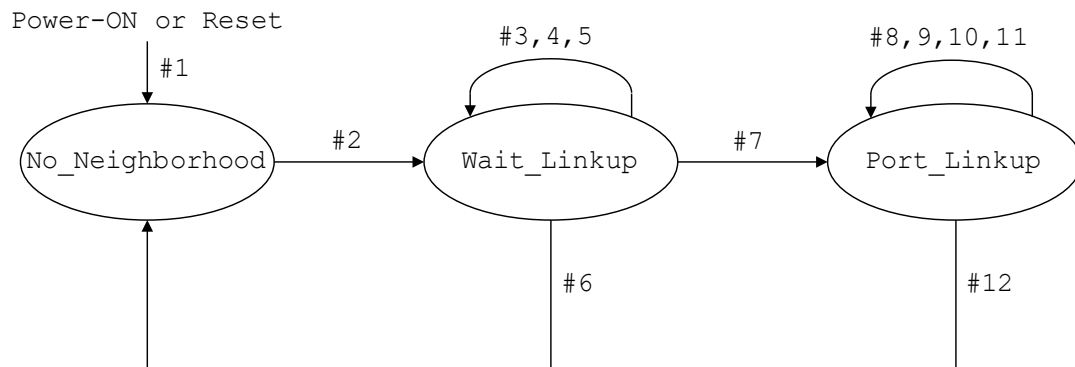


Figure 10 – State transition diagram of RHE_SM-A

Table 22 – Transitions of RHE_SM-A at RCL communication

| # | Current state | Event /condition => actions | Next state |
|---|-----------------|--|-----------------|
| 1 | Any states | Power-ON or Reset => V(PA_STATUS) = NNB V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL V(PA_NBST_RCLADD) = 0 C(PA_RHE_LKUP) = 0 V(PA_RHE_RxSeq) = 0 C(PA_RHE_RxCK) = 0 START_TIMER(T(PA_RHE_TIME), V(PA_RHE_CYCLE)) | No_Neighborhood |
| 2 | No_Neighborhood | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A => V(PA_STATUS) = WLU C(PA_RHE_LKUP) = 1 C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_NBST_RCLADD) = S_add V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |

| # | Current state | Event /condition => actions | Next state |
|---|---------------|--|-----------------|
| 3 | Wait_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && (V(PA_NBST_RCLADD) != S_add (V(PA_RHE_RxSeq)+1) != RHE_Seq) => C(PA_RHE_LKUP) = 1 C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_NBST_RCLADD) = S_add V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 4 | Wait_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && V(PA_NBST_RCLADD) == S_add && V(PA_RHE_RxSeq)+1 == RHE_Seq => C(PA_RHE_LKUP)++ C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 5 | Wait_Linkup | EXPIRED_TIMER(T(PA_RHE_TIME)) == "True" => C(PA_RHE_RxCK)-- | Wait_Linkup |
| 6 | Wait_Linkup | /C(PA_RHE_RxCK) == 0 => V(PA_STATUS) = NNB C(PA_RHE_LKUP) = 0 V(PA_NBST_RCLADD) = 0 V(PA_RHE_RxSeq) = 0 V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL | No_Neighborhood |
| 7 | Wait_Linkup | /C(PA_RHE_LKUP) == P(RHE_LKUP_NUM) => V(PA_STATUS) = PLU C(PA_Rx-RHE_NotMatch) = 0 V(PA_NBST_LKST) = UPDATE_LKST(DLSDU) V(PA_NBST_NDST) = UPDATE_NDST(DLSDU) | Port_Linkup |
| 8 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && V(PA_NBST_RCLADD) != S_add => C(PA_Rx-RHE_NotMatch)++ | Port_Linkup |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|--|-----------------|
| 9 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && V(PA_NBST_RCLADD) != S_add && C(PA_Rx- RHE_NotMatch) >= 1 => C(PA_Rx-RHE_NotMatch)++ C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_NBST_RCLADD) = S_add V(PA_NBST_LKST) = UPDATE_LKST(DLSDU) V(PA_NBST_NDST) = UPDATE_NDST(DLSDU) V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Port_Linkup |
| 10 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && V(PA_NBST_RCLADD) == S_add => C(PA_Rx-RHE_NotMatch) = 0 C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_NBST_LKST) = UPDATE_LKST(DLSDU) V(PA_NBST_NDST) = UPDATE_NDST(DLSDU) V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Port_Linkup |
| 11 | Port_Linkup | EXPIRED_TIMER(T(PA_RHEcycle)) == "True" => C(PA_RHE_RxCK)-- | Port_Linkup |
| 12 | Port_Linkup | /C(PA_RHE_RxCK) == 0 => V(PA_STATUS) = NNB C(PA_RHE_LKUP) = 0 V(PA_NBST_RCLADD) = 0 V(PA_RHE_RxSeq) = 0 V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL | No_Neighborhood |

7.2.3.1.2 State machine RHE_SM-B

Figure 11 shows the state diagram of RHE_SM-B, and Table 23 is the state table of port B.

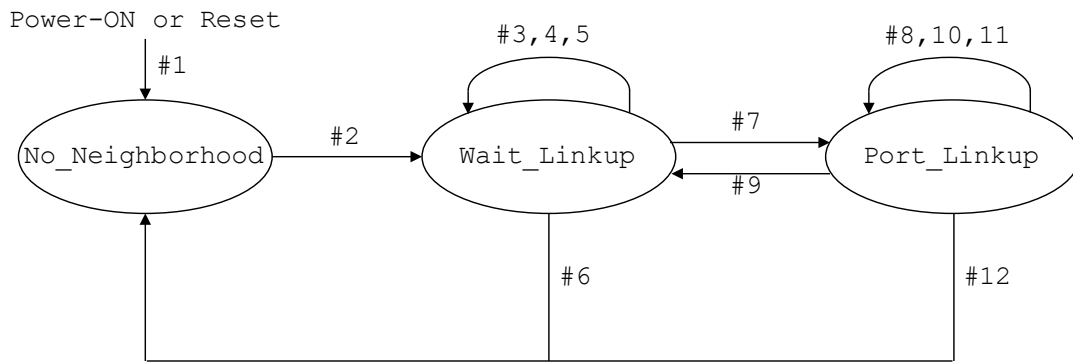


Figure 11 – State transition diagram of RHE_SM-B

Table 23 – Transitions of RHE_SM-B at RCL communication

| # | Current state | Event /condition => actions | Next state |
|---|-----------------|--|-----------------|
| 1 | Any states | Power-ON or Reset => V(PB_STATUS) = NNB V(PB_NBST_LKST) = NNB V(PB_NBST_NDST) = ISL V(PB_NBST_RCLADD) = 0 C(PB_RHE_LKUP) = 0 V(PB_RHE_RxSeq) = 0 C(PB_RHE_RxCK) = 0 START_TIMER(T(PB_RHE_TIME), V(PB_RHE_CYCLE)) | No_Neighborhood |
| 2 | No_Neighborhood | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B => V(PB_STATUS) = WLU C(PB_RHE_LKUP) = 1 C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_NBST_RCLADD) = S_add V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 3 | Wait_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && (V(PB_NBST_RCLADD) != S_add V(PB_RHE_RxSeq)+1 != RHE_Seq) => C(PB_RHE_LKUP) = 1 C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_NBST_RCLADD) = S_add V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |

| # | Current state | Event /condition => actions | Next state |
|---|---------------|---|-----------------|
| 4 | Wait_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && V(PB_NBST_RCLADD) == S_add && V(PB_RHE_RxSeq)+1 == RHE_Seq => C(PB_RHE_LKUP)++ C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 5 | Wait_Linkup | EXPIRED_TIMER(T(PB_RHE_TIME)) == "True" => C(PB_RHE_RxCK)-- | Wait_Linkup |
| 6 | Wait_Linkup | /C(PB_RHE_RxCK) == 0 => V(PB_STATUS) = NNB C(PB_RHE_LKUP) = 0 V(PB_NBST_RCLADD) = 0 V(PB_RHE_RxSeq) = 0 V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL | No_Neighborhood |
| 7 | Wait_Linkup | /C(PB_RHE_LKUP) == P(RHE_LKUP_NUM) => V(PB_STATUS) = PLU C(PB_Rx-RHE_NotMatch) = 0 V(PB_NBST_LKST) = UPDATE_LKST(DLSDU) V(PB_NBST_NDST) = UPDATE_NDST(DLSDU) | Port_Linkup |
| 8 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && V(PB_NBST_RCLADD) != S_add => C(PB_Rx-RHE_NotMatch)++ | Port_Linkup |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|---|-----------------|
| 9 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && V(PA_NBST_RCLADD) != S_add && C(PA_Rx-RHE_NotMatch) >= 1 => C(PB_Rx-RHE_NotMatch)++ C(PB_RHE_LKUP) = 1 C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_NBST_RCLADD) = S_add V(PB_NBST_LKST) = UPDATE_LKST(DLSDU) V(PB_NBST_NDST) = UPDATE_NDST(DLSDU) V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 10 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && V(NBST_RCLADD) == S_add => C(PB_Rx-RHE_NotMatch) = 0 C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_NBST_LKST) = UPDATE_LKST(DLSDU) V(PB_NBST_NDST) = UPDATE_NDST(DLSDU) V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Port_Linkup |
| 11 | Port_Linkup | EXPIRED_TIMER(T(PB_RHEcycle)) == "True" => C(PB_RHE_RxCK)-- | Port_Linkup |
| 12 | Port_Linkup | /C(PB_RHE_RxCK) == 0 => V(PB_STATUS) = NNB C(PB_RHE_LKUP) = 0 V(PB_NBST_RCLADD) = 0 V(PB_RHE_RxSeq) = 0 V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL | No_Neighborhood |

7.2.3.2 RCL node status state machine (RCLNode_SM)

The RCLNode_SM selects an Edge-A node and an Edge-B node in the network. All of the nodes in Type 25 network use this state machine to decide the status of their own node.

The node controls the link status of ports by RCL_SM-A and RCL_SM-B. The RCLNode_SM uses these link status and informs other nodes in the network by RCL frames. All nodes in the network select an Edge-A node and an Edge-B node using this state machine. This state machine has the following states.

- Node_ISL: Isolated node state, port A and B are blocked.

- Edge-A_PLD: Edge-A physical link down state. At the state, the node is the Edge-A node, port A is link up, and port B is blocked due to physical factor for example not connected to the cable.
- Edge-A_LLD: Edge-B logical link down state. At the state, the node is the Edge-A node, port A is link up, port B is blocked, and the ring network is normal.
- Edge-B_PLD: Edge-B physical link down state.
- Edge-B_LLD: Edge-B logical link down state.
- Node_ITM: Intermediate node state. Ports A and B are link up.

Figure 12 and Table 24 show the state diagram and the state table of RCLNode_SM.

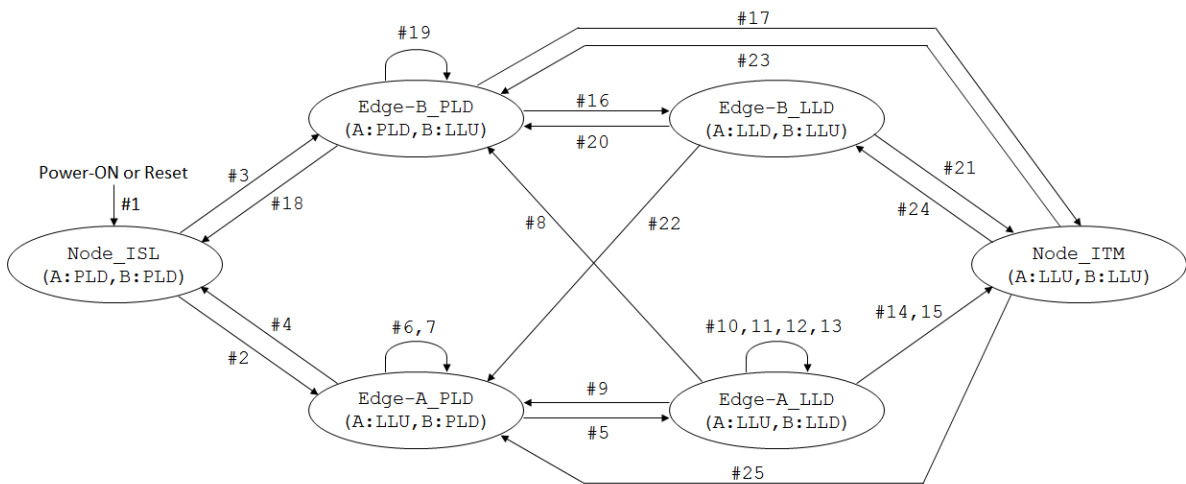


Figure 12 – The state diagram of RCLNode_SM

Table 24 – Transitions of RCLNode_SM at RCL communication

| # | Current state | Event /condition => actions | Next state |
|---|---------------|---|------------|
| 1 | Any states | Power-ON or Reset => Node_ST = ISL PortA_ST = PLD PortB_ST = PLD RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} BLKport = B RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ISL |

| # | Current state | Event /condition => actions | Next state |
|---|---------------|---|------------|
| 2 | Node_ISL | /V(PA_STATUS) == PLU => Node_ST = EGA PortA_ST = LLU BLKport = A Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_PLD |
| 3 | Node_ISL | /V(PB_STATUS) == PLU V(PB_NBST_NDST) != ISL => Node_ST = EGB PortB_ST = LLU BLKport = B Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_PLD |
| 4 | Edge-A_PLD | /V(PA_STATUS) == (NNB WLU) => RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} Node_ST = ISL PortA_ST = PLD BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ISL |
| 5 | Edge-A_PLD | /V(PB_STATUS) == PLU => RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} C(Rx_Counter) = 0 RCL_Indtype = LCC RCL_IndDA = Brdcast RCL_IndPri = Normal RCL_IndPort = Both RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} PortB_ST = LLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_LLD |

| # | Current state | Event /condition => actions | Next state |
|---|---------------|--|------------|
| 6 | Edge-A_PLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCC => RCL_Indtype = LCA RCL_IndDA = S_add RCL_IndPri = RCL_pri RCL_IndPort = Rx_port RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Edge-A_PLD |
| 7 | Edge-A_PLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LNA && D_add == MyS_add => RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} | Edge-A_PLD |
| 8 | Edge-A_LLD | /V(PA_STATUS) == (NNB WLU) => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} Node_ST = EGB PortA_ST = PLD BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} BLKport = B Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_PLD |
| 9 | Edge-A_LLD | /V(PB_STATUS) == (NNB WLU) => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} RCL_Indtype = LCN RCL_IndDA = Brdcast RCL_IndPri = V(LCN_Pri) RCL_IndPort = PortA RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} PortB_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_PLD |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|---|------------|
| 10 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCC && RCL_IndPri < RCL_Pri => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} C(Rx_Counter) = 0 RCL_IndDA = Brdcast RCL_IndPri = Normal RCL_IndPort = Both RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Edge-A_LLD |
| 11 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCC && RCL_IndPri > RCL_Pri => RCL_Indtype = LCA RCL_IndDA = S_add RCL_IndPri = RCL_pri RCL_IndPort = Rx_port RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} C(Rx_Counter) = 0 RCL_Indtype = LCC RCL_IndDA = Brdcast RCL_IndPri = Normal RCL_IndPort = Both RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Edge-A_LLD |
| 12 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCC && RCL_IndPri == RCL_Pri => C(Rx_Counter)++ | Edge-A_LLD |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|--|------------|
| 13 | Edge-A_LLD | /C(Rx_Counter) >= P(LCC_STOP_NUM) => C(Rx_Counter) = 0 RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_LLD |
| 14 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCA && D_add == MyS_add => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} BLKport = B Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST = ITM PortB_ST = LLU Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ITM |
| 15 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCN => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} BLKport = B Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST = ITM PortB_ST = LLU Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ITM |
| 16 | Edge-B_PLD | /V(PA_STATUS) == PLU && V(PA_NBST_NDST) == EGA => PortA_ST = LLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_LLD |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|--|------------|
| 17 | Edge-B_PLD | /V(PA_STATUS) == PLU && V(PA_NBST_NDST) == (EGB ITM) => BLKport = A Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST = ITM PortA_ST = LLU Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ITM |
| 18 | Edge-B_PLD | /V(PB_STATUS) == (NNB WLU) => RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} BLKport = B Blocking = True RCLC-SetPT.req{BLKport,Blocking} Node_ST = ISL PortB_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ISL |
| 19 | Edge-B_PLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCN => RCL_Indtype = LNA RCL_IndDA = S_add RCL_IndPri = V(LNA_Pri) RCL_IndPort = PortB RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Edge-B_PLD |
| 20 | Edge-B_LLD | /V(PA_STATUS) == (NNB WLU) => PortA_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_PLD |
| 21 | Edge-B_LLD | /V(PA_STATUS) == PLU && V(PA_NBST_NDST) == (EGB ITM) => BLKport = A Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST = ITM PortA_ST = LLU Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ITM |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|--|------------|
| 22 | Edge-B_LLD | /V(PB_STATUS) == (NNB WLU) => BLKport = B Blocking = True RCLC-SetPT.req{BLKport,Blocking} BLKport = A Blocking = False RCLC-SetPT.req{BLKport,Blocking} RCL_Indtype = LCN RCL_IndDA = Brdcast RCL_IndPri = V(LCN_pri) RCL_IndPort = PortA RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri} Node_ST = EGA PortA_ST = LLU PortB_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_PLD |
| 23 | Node_ITM | /V(PA_STATUS) == (NNB WLU) => BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} Node_ST = EGB PortA_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_PLD |
| 24 | Node_ITM | /V(PA_STATUS) == PLU && V(PA_NBST_NDST) == EGA => BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} Node_ST = EGB PortB_ST = LLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_LLD |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|---|------------|
| 25 | Node_ITM | /V(PB_STATUS) == (NNB WLU) => BLKport = B Blocking = True RCLC-SetPT.req{BLKport,Blocking} RCL_Indtype = LCN RCL_IndDA = Brdcast RCL_IndPri = V(LCN_pri) RCL_IndPort = PortA RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} Node_ST = EGA PortB_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_PLD |

7.2.3.3 RCL Transmit Receive state machine (RCLTR_SM)

The RCLTR_SM performs transit functions between the DLS-user and the TRC.

Figure 13 shows the state diagram of RCLT_SM, and Table 25 is the state table of RCLTR_SM.

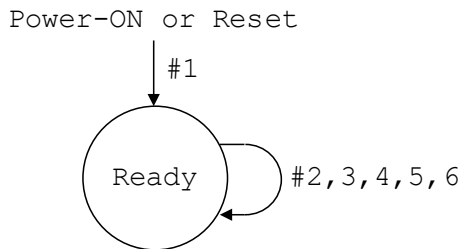


Figure 13 – The state diagram of RCLTR_SM

Table 25 – Transitions of RCLTR_SM at RCL communication

| # | Current state | Event /condition => actions | Next state |
|---|---------------|---|------------|
| 1 | Any state | Power-ON or Reset => | Ready |
| 2 | Ready | DL-RCL.req{S_add,PortNum,Frame_pri, RCL_type,DLSDU} /RCL_type == RHE && CHK_TRCST() == "True" => Status = "Success" D_add = Class1_address Tx_port = PortNum RCLC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} DL-RCL.cnf{Status} | Ready |
| 3 | Ready | DL-RCL.req{S_add,PortNum,Frame_pri, RCL_type,DLSDU} /RCL_type == RHE && CHK_TRCST() != "True" => Status = "Failure" DL-RCL.cnf{Status} | Ready |
| 4 | Ready | DL-RCL.req{S_add,PortNum,Frame_pri, RCL_type,DLSDU} /RCL_type == (LCC LCA LCN LNA SCR) && CHK_TRCST() == "True" => Status = "Success" D_add = Class2_address Tx_port = PortNum RCLC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} DL-RCL.cnf{Status} | Ready |
| 5 | Ready | DL-RCL.req{S_add,PortNum,Frame_pri, RCL_type,DLSDU} /RCL_type == (LCC LCA LCN LNA SCR) && CHK_TRCST() != "True" => Status = "Failure" DL-RCL.cnf{Status} | Ready |

| # | Current state | Event /condition => actions | Next state |
|---|---------------|--|------------|
| 6 | Ready | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_Pri,RCL_type,DLSDU} => PortNum = Rx_port DL-RCL.ind{S_add,PortNum,Frame_pri, RCL_type,DLSDU} | Ready |

7.2.4 Function of RCLC

All the functions used by the RCLC are summarized in Table 26.

Table 26 – RCLC function table

| Function name | Input | Output | Description and operation |
|---------------|-------------|-------------------------------|--|
| UPDATE_LKST | DLSDU | Link Status (NNB/WLU/PLU) | Return link status of the neighboring node. The link status is included in the received DLSDU. |
| UPDATE_NDST | DLSDU | Node Status (ISL/EGA/EGB/ITM) | Return node status of the neighboring node. The node status is included in the received DLSDU. |
| UPDATE_RHESeq | DLSDU | 0x00000000 ~ 0xFFFFFFFF | Return 4 octets of sequence number contained in DLSDU. |
| START_TIMER | TIM_ID, Val | (<none>) | Timer TIM_ID is set by value of Val and activated. |
| EXPIRED_TIMER | TIM_ID | True/False | When the requested timer TIM_ID has expired, "True" is returned, otherwise False is returned. |
| CHK_TRCST | (<none>) | True/False | Return status of TRC. If TRC is able to send the frame due to absence of other frames in transmit buffer, it returns "True," otherwise it returns "False." |

7.3 Real-time communication control (RTC)

7.3.1 General

The RTC provides to send and receive the cyclic, control, and information communication frames.

7.3.2 Primitive definitions

7.3.2.1 Primitive definitions between RTC and DLS-user

Table 27 summarizes all primitives exchanged between the RTC and the DLS-user.

Table 27 – The primitives and parameters for DLS-user interface

| Primitive name | Source | Associated parameters | Description |
|----------------|----------|---|--|
| DL-RTC.req | DLS-user | (in D_add, S_add, Frame_pri, DLSDU) | Transmits a RT frame from the DLS-user. The RT frame is carried to TRC with associated parameters. |
| DL-RTC.cnf | RTC | (out Status) | Responds to complete DLSDU transmission and reports the status to DLS-user. |
| DL-RTC.ind | RTC | (out D_add, S_add, Frame_pri, DLSDU) | Carries forward a received RT frame to the DLS-user. |

The parameters used with the primitives exchange between the RTC and the DLS-user are described in Table 28.

Table 28 – Parameters used with primitives exchanged between RTC and DLS-user

| Parameter name | Description |
|----------------|--|
| D_add | The D_add parameter specifies the DL-address of the subscriber. |
| S_add | The S_add parameter specifies the DL-address of the publisher. |
| Frame_pri | This parameter specifies the priority with VLAN of the ring network. |
| DLSDU | This parameter specifies the information of the RT frame. |
| Status | This parameter allows DLMS-user to determine whether the requested DLS was provided successfully, or failed due to a particular reason. The value conveyed in this parameter is as follows: "OK – successfully completed"; "Failure – terminated before completion". |

7.3.2.2 Primitive definitions between RTC and TRC

Table 29 summarizes all primitives exchanged between the RTC and the TRC.

Table 29 – The primitives and parameters for TRC interface

| Primitive name | Source | Associated parameters | Description |
|----------------|--------|---|--|
| RTC-Tx.req | RTC | D_add, S_add, Tx_port Frame_pri, DLSDU | Transmits a RT frame to TRC with associated parameters. The RTC attaches the Tx_port "Both" parameter. |
| RTC-Rx.ind | TRC | D_add, S_add, Rx_port, Frame_pri, DLSDU | Carries forward a received RT frame from the TRC. |

The parameters used with the primitives exchange between the RTC and the TRC are described in Table 30.

Table 30 – Parameters used with primitives exchanged between RTC and TRC

| Parameter name | Description |
|----------------|--|
| D_add | The D_add parameter specifies the DL-address of the subscriber. |
| S_add | The S_add parameter specifies the DL-address of the publisher. |
| Tx_port | This parameter specifies the transmit port of the publisher. |
| Rx_port | This parameter specifies the receive port of the publisher. |
| Frame_pri | This parameter specifies the priority with VLAN of the ring network. |
| DLSDU | This parameter specifies the information of the RT frame. |

7.3.2.3 Primitive definitions between RTC and DLM

Table 31 summarizes all primitives exchanged between the RTC and the DLM.

Table 31 – The primitives and parameters for DLM interface

| Primitive name | Source | Associated parameters | Description |
|----------------|--------|---|---|
| RTC-Reset.ind | DLM | (none) | Indicates a reset of the RTC from the DLS-user. |
| RTC-Event.ind | RTC | DLM_event identifier, Additional_information | This service is used to inform the DLM about certain events or errors in the DLL. |

The parameters used with the primitives exchange between the RTC and the DLM are described in Table 32.

Table 32 – Parameters used with primitives exchanged between RTC and DLM

| Parameter name | Description |
|------------------------|---|
| DLM_event identifier | This parameter specifies the primitive or composite event within the DLE whose occurrence is being announced. |
| Additional_information | This optional parameter provides event-specific additional information. |

7.3.3 RTC state machine

7.3.3.1 RT Transmit Receive state machine (RTTR_SM)

The RTTR_SM performs transit functions between the DLS-user and the TRC.

Figure 14 shows the state diagram of RTTR_SM, and Table 33 is the state table of RTTR_SM.

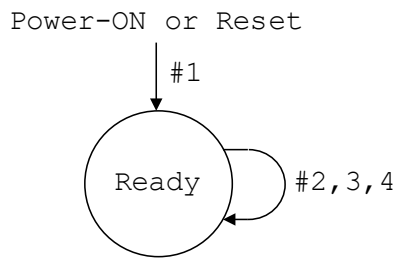


Figure 14 – The state diagram of RTTR_SM

Table 33 – Transitions of RTTR_SM at RT communication

| # | Current state | Event /condition => actions | Next state |
|---|---------------|---|------------|
| 1 | Any state | Power-ON or Reset => | Ready |
| 2 | Ready | DL-RTC.req{D_add,S_add,Frame_pri,DLSDU} /CHK_TRCST() == "True" => Status = "Success" Tx_port = Both RTC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} DL-RTC.cnf{Status} | Ready |
| 3 | Ready | DL-RTC.req{D_add,S_add,Frame_pri,DLSDU} /CHK_TRCST() != "True" => Status = "Failure" DL-RTC.cnf{Status} | Ready |
| 4 | Ready | RTC-Rx.ind{D_add,S_add,Rx_port, Frame_Pri,DLSDU} => DL-RTC.ind{D_add,S_add,Frame_pri,DLSDU} | Ready |

7.3.4 Function of RTC

All the functions used by the RTC are summarized in Table 34.

Table 34 – RTC function table

| Function name | Input | Output | Description and operation |
|---------------|----------|------------|---|
| CHK_TRCST | (<none>) | True/False | Returns status of TRC. If TRC is able to send the frame due to without other frames in transmit buffer, it returns "True," otherwise returns "False." |

7.4 Transmit/Receive control (TRC)

7.4.1 General

The TRC provides the following functions:

- Send and receive all the frames in Type 25 network;
- Control the port status at the instruction of RCLC and indicate the port status to the RCLC;
- Transit the received frames from a port to another port.

7.4.2 Primitive definitions

7.4.2.1 Primitive definitions between TRC and DLM

Table 35 summarizes all primitives exchanged between the TRC and the DLM.

Table 35 – The primitives and parameters for DLM interface

| Primitive name | Source | Associated parameters | Description |
|----------------|--------|---|---|
| TRC-Reset.ind | DLM | (none) | Indicates a reset of the TRC from the DLS-user. |
| TRC-Event.ind | TRC | DLM_event identifier, Additional_information | This service is used to inform the DLM about certain events or errors in the DLL. |

The parameters used with the primitives exchange between the TRC and the DLM are described in Table 36.

Table 36 – Parameters used with primitives exchanged between TRC and DLM

| Parameter name | Description |
|------------------------|---|
| DLM_event identifier | This parameter specifies the primitive or composite event within the DLE whose occurrence is being announced. |
| Additional_information | This optional parameter provides event-specific additional information. |

7.4.3 TRC state machine

7.4.3.1 TRC state machine (TRC_SM)

The TRC_SM controls whether to send or to receive the RCL and RT frames according to the basis of blocking status determined by RCLC. The TRC_SM indicates the port status to the RCLC.

Figure 15 shows the state diagram of TRC_SM, and Table 37 is the state table of TRC_SM.

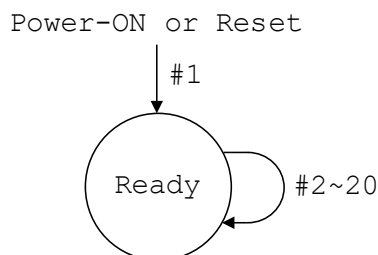


Figure 15 – The state diagram of TRC_SM

Table 37 – Transitions of TRC_SM

| # | Current state | Event /condition => actions | Next state |
|---|---------------|---|------------|
| 1 | Any state | Power-ON or Reset => V(PA_Blocking) = True V(PB_Blocking) = True | Ready |
| 2 | Ready | RCLC-SetPT.req{BLKport,Blocking} /BLKport == A => V(PA_Blocking) = Blocking | Ready |
| 3 | Ready | RCLC-SetPT.req{BLKport,Blocking} /BLKport == B => V(PB_Blocking) = Blocking | Ready |
| 4 | Ready | Phs-A_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add == Class1frame => Rx_port = A RCL_type = GET_RCLTYPE(DLSDU) RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} | Ready |
| 5 | Ready | Phs-B_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add == Class1frame => Rx_port = B RCL_type = GET_RCLTYPE(DLSDU) RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} | Ready |

| # | Current state | Event /condition => actions | Next state |
|---|---------------|--|------------|
| 6 | Ready | <pre> Phs-A_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add == Class2frame && !(V(PA_Blocking) == True && V(PB_Blocking) == True) => Rx_port = A RCL_type = GET_RCLTYPE(DLSDU) RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} IF(V(PA_Blocking) == True V(PB_Blocking) == False) THEN Put_Queue-B(D_add,S_add,Frame_pri, DLSDU) ENDIF </pre> | Ready |
| 7 | Ready | <pre> Phs-B_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add == Class2frame && !(V(PA_Blocking) == True && V(PB_Blocking) == True) => Rx_port = B RCL_type = GET_RCLTYPE(DLSDU) RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} IF(V(PA_Blocking) == True V(PB_Blocking) == False) THEN Put_Queue-A(D_add,S_add,Frame_pri, DLSDU) ENDIF </pre> | Ready |
| 8 | Ready | <pre> Phs-A_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add != Class1frame && D_add != Class2frame && V(PA_Blocking) == False => Rx_port = A RLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,DLSDU} IF(V(PB_Blocking) == False) THEN Put_Queue-B(D_add,S_add,Frame_pri, DLSDU) ENDIF </pre> | Ready |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|--|------------|
| 9 | Ready | <pre> Phs-B_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add != Class1frame && D_add != Class2frame && V(PB_Blocking) == False => Rx_port = B RLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,DLSDU} IF(V(PA_Blocking) == False) THEN Put_Queue-A(D_add,S_add,Frame_pri, DLSDU) ENDIF </pre> | Ready |
| 10 | Ready | <pre> RCLC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} /D_add == Class1frame => IF(Tx_port == A) THEN Put_Queue-A(D_add,S_add,Frame_pri, DLSDU) ENDIF IF(Tx_port == B) THEN PUT_Queue-B(D_add,S_add,Frame_pri, DLSDU) ENDIF </pre> | Ready |
| 11 | Ready | <pre> RCLC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} /D_add == Class2frame && !(V(PA_Blocking) == True && V(PB_Blocking) == True) => IF(Tx_port == Both Tx_port == A) THEN Put_Queue-A(D_add,S_add,Frame_pri, DLSDU) IF(Tx_port == Both Tx_port == B) THEN Put_Queue-B(D_add,S_add,Frame_pri, DLSDU) </pre> | Ready |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|--|------------|
| 12 | Ready | RTC-Tx.req{D_add,S_add,Tx_port, Frame_pri,DLSDU} /D_add != Class1frame && D_add != Class2frame && Tx_port == Both => IF(V(PA_Blocking) == False) THEN Put_Queue-A(D_add,S_add, Frame_pri,DLSDU) ENDIF IF(V(PB_Blocking) == False) THEN Put_Queue-B(D_add,S_add, Frame_pri,DLSDU) ENDIF | Ready |
| 13 | Ready | /Queue-A_Check(VLAN_RCL) == True => Send_Quuedata-A(VLAN_RCL) | Ready |
| 14 | Ready | /Queue-B_Check(VLAN_RCL) == True => Send_Quuedata-B(VLAN_RCL) | Ready |
| 15 | Ready | /Queue-A_Check(VLAN_CYC) == True && Queue-A_Check(VLAN_RCL) == False => Send_Quuedata-A(VLAN_CYC) | Ready |
| 16 | Ready | /Queue-B_Check(VLAN_CYC) == True && Queue-B_Check(VLAN_RCL) == False => Send_Quuedata-B(VLAN_CYC) | Ready |
| 17 | Ready | /Queue-A_Check(VLAN_CTL) == True && Queue-A_Check(VLAN_RCL) == False && Queue-A_Check(VLAN_CYC) == False => Send_Quuedata-A(VLAN_CTL) | Ready |
| 18 | Ready | /Queue-B_Check(VLAN_CTL) == True && Queue-B_Check(VLAN_RCL) == False && Queue-B_Check(VLAN_CYC) == False => Send_Quuedata-B(VLAN_CTL) | Ready |

| # | Current state | Event /condition => actions | Next state |
|----|---------------|--|------------|
| 19 | Ready | /Queue-A_Check(VLAN_INFO) == True && Queue-A_Check(VLAN_RCL) == False && Queue-A_Check(VLAN_CYC) == False && Queue-A_Check(VLAN_CTL) == False => Send_Queueudata-A(VLAN_INFO) | Ready |
| 20 | Ready | /Queue-B_Check(VLAN_INFO) == True && Queue-B_Check(VLAN_RCL) == False && Queue-B_Check(VLAN_CYC) == False && Queue-B_Check(VLAN_CTL) == False => Send_Queueudata-B(VLAN_INFO) | Ready |

7.4.4 Function of TRC

All the functions used by the TRC are summarized in Table 38.

Table 38 – TRC function table

| Function name | Input | Output | Description and operation |
|-------------------|---|---|---|
| GET_RCLTYPE | DLSDU | RCL_type (RHE/LCC/ LCA/LCN/ LNA/SCR) | Returns the RCL_type included in DLSDU. |
| Put_Queue-A | D_add, S_add, Frame_pri, DLSDU | (<none>) | Queues the input PDU into the port A transmit Queue appropriate to Frame_pri on a FIFO basis. |
| Put_Queue-B | D_add, S_add, Frame_pri, DLSDU | (<none>) | Queues the input PDU into the port B transmit Queue appropriate to Frame_pri on a FIFO basis. |
| Queue-A_Check | Frame_pri | True/False | Returns the “True” if the queue of port A specified by Frame_pri is not empty. |
| Queue-B_Check | Frame_pri | True/False | Returns the “True” if the queue of port B specified by Frame_pri is not empty. |
| Send_Queueudata-A | Frame_pri | (<none>) | Dequeues from the queue of port A specified by Frame_pri on a FIFO basis and sends the PDU from the port A. Send_Queueudata-A(Frame_pri) is assembled as follows: PortNum = PortA D_add = dequeue_Dadd(PortNum, Frame_Pri) S_add = dequeue_Sadd(PortNum, Frame_Pri) DLSDU = dequeue_data(PortNum, Frame_Pri) Phs-A_Data.req{D_add,S_add, Frame_pri,DLSDU} |

| Function name | Input | Output | Description and operation |
|------------------|-----------------------|----------|--|
| Send_QueueData-B | Frame_pri | (<none>) | Dequeues from the queue of port A specified by Frame_pri on a FIFO basis and sends the PDU from port A. Send_QueueData-B(Frame_pri) is assembled as follows: PortNum = PortB D_add = queue_Dadd(PortNum, Frame_Pri) S_add = queue_Sadd(PortNum, Frame_Pri) DLSDU = queue_data(PortNum, Frame_Pri) Phs-B_Data.req{D_add,S_add, Frame_pri,DLSDU} |
| dequeue_Dadd | PortNum, Frame_Pri | D_add | Dequeues the destination address from the queue specified by PortNum and Frame_Pri on a FIFO basis. |
| dequeue_Sadd | PortNum, Frame_Pri | S_add | Dequeues the source address from the queue specified by PortNum and Frame_Pri on a FIFO basis. |
| dequeue_data | PortNum, Frame_Pri | DLSDU | Dequeues the DLSDU from the queue specified by PortNum and Frame_Pri on a FIFO basis. |

7.5 DLL management protocol (DLM)

7.5.1 Overview

The interface protocol between the DLM and the DLS-user is described in 7.5.

7.5.2 Primitive definitions

7.5.2.1 Primitive exchanged between DLM and DLS-user

Table 39 summarizes all primitives exchanged between the DLM and the DLS-user.

Table 39 – Primitives exchanged between DLM and DLS-user

| Primitive name | Source | Associated parameters | Description |
|-------------------|--------|--|--|
| DLM_RCL_STOP.ind | DLM | (out RCL_Indtype) | Indicates a stop of the RCL frame to DLS-user based on the ring control state machine. |
| DLM_RCL_START.ind | DLM | (out RCL_Indtype, RCL_IndDA, RCL_IndPri, RCL_IndPort) | Indicates a start of the RCL frame to DLS-user based on the ring control state machine. |
| DLM_Node_ST.ind | DLM | (out Node_ST, PortA_ST, PortB_ST) | Indicates a change of the node or port status to DLS-user based on the ring control state machine. |

The parameters used with the primitives exchanged between the DLM and the DLS-user are described in Table 40.

Table 40 – Parameters used with primitives exchanged between DLM and DLS-user

| Parameter name | Description |
|----------------|---|
| RCL_Indtype | This parameter specifies the RCL type of the requested RCL frames for the DLM due to the change of the node status. |
| RCL_IndDA | This parameter specifies the DL-address of the requested RCL frames for the DLM due to the change of the node status. |
| RCL_IndPri | This parameter specifies the priority with VLAN of the requested RCL frames for the DLM due to the change of the node status. |
| RCL_IndPort | This parameter specifies the transmission port of the requested RCL frames for the DLM due to the change of the node status. |
| Node_ST | This parameter contains the node status (ISL/EGA/EGB/ITM) for the DLM due to the change of the node status. |
| PortA_ST | This parameter contains the status of port A on own node for the DLM due to the change of the node status. |
| PortB_ST | This parameter contains the status of port B on own node for the DLM due to the change of the node status. |

7.5.3 DLM state machine (DLM_SM)

Figure 16 shows the state diagram of DLM_SM, and Table 41 is the state table of DLM_SM.

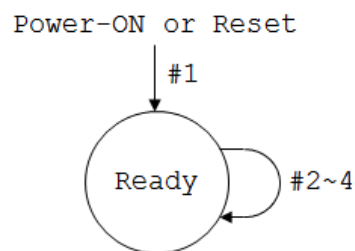


Figure 16 – The state diagram of DLM_SM

Table 41 – Transitions of DLM_SM

| # | Current state | Event /condition => actions | Next state |
|---|---------------|---|------------|
| 1 | Any state | Power-ON or Reset => | Ready |
| 2 | Ready | RCL_STOP.ind{RCL_Indtype} / => DLM_RCL_STOP.ind{RCL_Indtype} | Ready |
| 3 | Ready | RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} / => DLM_RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Ready |
| 4 | Ready | Node_ST.ind{Node_ST,PortA_ST,PortB_ST} / => DLM_Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Ready |

Bibliography

IEC 61158-1:2019, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

SOMMAIRE

| | |
|---|----|
| AVANT-PROPOS..... | 65 |
| INTRODUCTION..... | 67 |
| 1 Domaine d'application | 69 |
| 1.1 Généralités | 69 |
| 1.2 Spécifications | 69 |
| 1.3 Procédures | 69 |
| 1.4 Applicabilité | 70 |
| 1.5 Conformité | 70 |
| 2 Références normatives | 70 |
| 3 Termes, définitions, symboles, abréviations et conventions | 70 |
| 3.1 Termes et définitions du modèle de référence..... | 71 |
| 3.2 Termes et définitions de convention de service..... | 72 |
| 3.3 Termes et définitions | 73 |
| 3.4 Symboles et abréviations | 75 |
| 3.5 Conventions communes | 76 |
| 3.6 Conventions de type 25 supplémentaires | 78 |
| 3.6.1 Conventions pour les primitives | 78 |
| 3.6.2 Conventions dans les diagrammes d'états | 78 |
| 4 Vue d'ensemble du DL-protocol | 79 |
| 4.1 Généralités | 79 |
| 4.2 Vue d'ensemble du contrôle d'accès au support physique..... | 79 |
| 4.2.1 Généralités | 79 |
| 4.2.2 La topologie du réseau | 79 |
| 4.2.3 Contrôle de priorité avec VLAN..... | 82 |
| 4.2.4 Retard de remise maximal dans le réseau de Type 25..... | 83 |
| 4.2.5 Commande de trafic pour la communication en temps réel | 85 |
| 4.3 Service pris en charge à partir de la PhL | 86 |
| 4.4 Architecture de la couche DL | 86 |
| 4.5 Paramètres et variables locaux | 88 |
| 4.5.1 Vue d'ensemble..... | 88 |
| 4.5.2 Variables, paramètres, compteur et temporisateur | 89 |
| 5 Structure générale et codage des PHIDU et DLPDU, mais aussi éléments de procédure connexes | 90 |
| 5.1 Vue d'ensemble | 90 |
| 5.2 Structure, codage et éléments de trame MAC commune | 90 |
| 5.2.1 Structure de trame MAC | 90 |
| 5.2.2 Eléments de la trame MAC | 91 |
| 6 Structure spécifique à la DLPDU, codage et éléments de procédure..... | 93 |
| 6.1 Généralités | 93 |
| 6.2 Structure de la DLPDU de RCL | 94 |
| 6.2.1 En-tête RCL..... | 94 |
| 7 Eléments de procédure de DLE | 96 |
| 7.1 Vue d'ensemble | 96 |
| 7.2 Commande de communication RCL (RCLC)..... | 96 |
| 7.2.1 Généralités | 96 |
| 7.2.2 Définitions de primitives..... | 96 |

| | | |
|-------|---|-----|
| 7.2.3 | Diagramme d'états RCLC | 99 |
| 7.2.4 | Fonctions de la RCLC | 116 |
| 7.3 | Commande de communication en temps réel (RTC) | 116 |
| 7.3.1 | Généralités | 116 |
| 7.3.2 | Définitions de primitives | 116 |
| 7.3.3 | Diagramme d'états RTC | 118 |
| 7.3.4 | Fonctions de la RTC | 119 |
| 7.4 | Commande de transmission/réception (TRC) | 120 |
| 7.4.1 | Généralités | 120 |
| 7.4.2 | Définitions de primitives | 120 |
| 7.4.3 | Diagramme d'états TRC | 120 |
| 7.4.4 | Fonctions de la TRC | 125 |
| 7.5 | Protocole de gestion DLL (DLM) | 126 |
| 7.5.1 | Vue d'ensemble | 126 |
| 7.5.2 | Définitions de primitives | 126 |
| 7.5.3 | Diagramme d'états DLM (DLM_SM) | 127 |
| | Bibliographie | 129 |
| | | |
| | Figure 1 – Relations entre DLSAP, adresses de DLSAP et adresses de DL de groupe | 77 |
| | Figure 2 – Commande d'anneau dans le réseau de Type 25 | 80 |
| | Figure 3 – Plages de communication des trames de Type 25 | 82 |
| | Figure 4 – Contrôle de priorité avec VLAN de réseau de Type 25 | 83 |
| | Figure 5 – Mécanisme de retard de transmission d'un nœud | 84 |
| | Figure 6 – Retard maximal du réseau de Type 25 | 86 |
| | Figure 7 – Architecture interne de la couche de liaison de données | 88 |
| | Figure 8 – TFormat de la trame de DLPDU de bus de terrain de Type 25 | 91 |
| | Figure 9 – Format de trame RCL | 93 |
| | Figure 10 – Schéma de transition d'états de RHE_SM-A | 100 |
| | Figure 11 – Schéma de transition d'états de RHE_SM- B | 103 |
| | Figure 12 – Diagramme d'états de RCLNode_SM | 107 |
| | Figure 13 – Diagramme d'états de RCLTR_SM | 114 |
| | Figure 14 – Diagramme d'états de RTTR_SM | 119 |
| | Figure 15 – Diagramme d'états de TRC_SM | 121 |
| | Figure 16 – Diagramme d'états de DLM_SM | 127 |
| | | |
| | Tableau 1 – Descriptions des transitions d'état | 78 |
| | Tableau 2 – Description des éléments d'un diagramme d'états | 78 |
| | Tableau 3 – Conventions utilisées dans les diagrammes d'états | 79 |
| | Tableau 4 – Caractéristiques des états du nœud | 81 |
| | Tableau 5 – Caractéristique des classes de trame | 81 |
| | Tableau 6 – Mapping de priorité VLAN du réseau de Type 25 | 83 |
| | Tableau 7 – Composants de la couche de liaison de données | 87 |
| | Tableau 8 – Format de l'adresse de destination | 92 |
| | Tableau 9 – Format de l'étiquette VLAN | 92 |
| | Tableau 10 – Types et classes des trames RCL | 94 |

| | |
|--|-----|
| Tableau 11 – Structure de l'en-tête RCL | 94 |
| Tableau 12 – Format du champ | 95 |
| Tableau 13 – Format du champ adresse de destination | 95 |
| Tableau 14 – Format du champ Adresse de source..... | 95 |
| Tableau 15 – Format du champ Adresse de source..... | 96 |
| Tableau 16 – Primitives et paramètres de l'interface d'utilisateur DLS..... | 97 |
| Tableau 17 – Paramètres utilisés avec des primitives échangées entre DLM et utilisateur DLS | 97 |
| Tableau 18 – Primitives et paramètres de l'interface d'utilisateur TRC | 98 |
| Tableau 19 – Paramètres utilisés avec des primitives échangées entre la RCLC et la TRC... | 98 |
| Tableau 20 – Primitives et paramètres de l'interface DLM..... | 99 |
| Tableau 21 – Paramètres utilisés avec des primitives échangées entre DLPM et DLM..... | 99 |
| Tableau 22 – Transitions de RHE_SM-A au niveau de la communication RCL | 101 |
| Tableau 23 – Transitions de RHE_SM-B au niveau de la communication RCL | 104 |
| Tableau 24 – Transitions de RCLNode_SM au niveau de la communication RCL | 107 |
| Tableau 25 – Transitions de RCLTR_SM au niveau de la communication RCL..... | 115 |
| Tableau 26 – Table des fonctions de la RCLC | 116 |
| Tableau 27 – Primitives et paramètres de l'interface d'utilisateur DLS..... | 117 |
| Tableau 28 – Paramètres utilisés avec des primitives échangées entre DLM et utilisateur DLS | 117 |
| Tableau 29 – Primitives et paramètres de l'interface d'utilisateur TRC | 117 |
| Tableau 30 – Paramètres utilisés avec des primitives échangées entre la RTC et la TRC... | 118 |
| Tableau 31 – Primitives et paramètres de l'interface DLM..... | 118 |
| Tableau 32 – Paramètres utilisés avec des primitives échangées entre DLPM et DLM..... | 118 |
| Tableau 33 – Transitions de RTTR_SM au niveau de la communication RT | 119 |
| Tableau 34 – Table des fonctions RTC | 120 |
| Tableau 35 – Primitives et paramètres de l'interface DLM..... | 120 |
| Tableau 36 – Paramètres utilisés avec des primitives échangées entre DLPM et DLM..... | 120 |
| Tableau 37 – Transitions de TRC_SM..... | 121 |
| Tableau 38 – Table des fonctions de la TRC..... | 125 |
| Tableau 39 – Primitives échangées entre DLM et utilisateur DLS..... | 126 |
| Tableau 40 – Paramètres utilisés avec des primitives échangées entre DLM et utilisateur DLS | 127 |
| Tableau 41 – Transitions de DLM_SM | 128 |

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**RÉSEAUX DE COMMUNICATION INDUSTRIELS –
SPÉCIFICATIONS DES BUS DE TERRAIN –****Partie 4-25: Spécification du protocole de la couche liaison de données –
Éléments de type 25**

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC 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, l'IEC – 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 l'IEC"). 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 l'IEC, participent également aux travaux. L'IEC 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 l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC 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 l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC 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 l'IEC. L'IEC 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 l'IEC, 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 l'IEC, 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 l'IEC ou de toute autre Publication de l'IEC, 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.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 61158-4-25 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

La présente version bilingue (2020-12) correspond à la version anglaise monolingue publiée en 2019-04.

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

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

Une liste de toutes les parties de la série IEC 61158, publiées sous le titre général *Réseaux de communications industriels – Spécifications des bus de terrain*, est disponible sur le site web de l'IEC.

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 l'IEC 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.

INTRODUCTION

La présente partie de l'IEC 61158 est l'une d'une série produite pour faciliter l'interconnexion des composants d'un système d'automatisation. Elle est liée à d'autres normes de la série telle que définie par le modèle de référence des bus de terrain "à trois couches" décrit dans l'IEC 61158-1.

Le protocole de liaison de données assure un service de liaison de données en s'appuyant sur les services offerts par la couche physique. Le présent document a pour principal objet de fournir un ensemble de règles de communication, exprimées sous la forme des procédures que doivent réaliser des entités de liaison de données homologues (DLE) au moment de la communication. Ces règles pour la communication visent à fournir une base solide pour le développement et de servir une diversité de besoins:

- a) guider les implémenteurs et les concepteurs;
- b) pour une utilisation dans les essais et achats d'équipements;
- c) comme partie intégrante d'un accord pour l'admission de systèmes dans l'environnement de systèmes ouverts;
- d) comme affinement pour la compréhension de communications prioritaires au sein de l'OSI (Open Systems Interconnexion, c'est-à-dire Interconnexion des systèmes ouverts).

Cette norme traite, en particulier, de la communication et de l'interfonctionnement des capteurs, effecteurs et autres appareils d'automatisation. L'utilisation conjointe du présent document avec d'autres normes entrant dans les modèles de référence OSI ou de bus de terrain permet à des systèmes qui ne le pourraient pas sans cela, de fonctionner ensemble dans toute combinaison.

NOTE L'utilisation de certains des types de protocoles associés est limitée par leurs détenteurs de droits de propriété intellectuelle. Dans tous les cas, l'engagement visant à limiter l'abandon des droits de propriété intellectuelle prévus par les détenteurs de ces droits permet d'utiliser un type de protocole de couche de liaison de données particulier avec les protocoles de couche physique et de couche d'application dans les combinaisons de type, comme spécifié explicitement dans les parties relatives au profil. L'utilisation de différents types de protocole dans d'autres combinaisons peut impliquer d'obtenir l'autorisation auprès de leurs détenteurs de droit de propriété intellectuelle respectifs.

La commission électrotechnique internationale (IEC) attire l'attention sur le fait qu'il est déclaré que la conformité avec les dispositions du présent document peut impliquer l'utilisation de brevets concernant les éléments de Type 25 et éventuellement d'autres types donnés dans le présent document comme suit:

| | |
|------------------|--|
| JP4074631 [HI] | Système de lignes de transmission, et émetteur de trame intégré dans le système et procédé de commutation de ligne de transmission |
| JP4653800 [HI] | Système de lignes de transmission, appareillage de transmission de trames, procédé et programme pour la commutation de ligne de transmission dans le système de lignes de transmission |
| JP4944986 [HI] | Système de lignes de transmission et procédé de construction de ligne de transmission |
| CN1964307 [HI] | Système de chemins de transfert et dispositif de transfert de trame intégré dans le système, procédé et système de transfert de chemin de transfert |
| CN101515887 [HI] | Système de lignes de transmission, et émetteur de trame intégré dans le système, procédé et programme de commutation de ligne de transmission |

L'IEC ne prend pas position concernant la preuve, la validité et le domaine d'application de ce droit de propriété intellectuelle.

Le détenteur de ce droit de propriété intellectuelle a assuré à l'IEC qu'il souhaitait négocier des licences avec des demandeurs situés dans le monde entier, soit gratuitement, soit dans des termes et conditions raisonnables et non discriminatoires. A cet égard, la déclaration du détenteur de ce droit de propriété intellectuelle est enregistrée auprès de l'IEC. Des informations peuvent être obtenues en écrivant à l'adresse suivante:

[HI] Hitachi, Ltd.
Intellectual Property Division (Division de propriété intellectuelle)
20-2, Saiwai-cho 1-chome, Hitachi-shi,
Ibaraki-ken, 317-0073 Japon

L'attention est attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété intellectuelle autres que ceux qui sont identifiés ci-dessus. L'IEC ne doit pas être tenue responsable de l'identification de tout ou partie de ces droits de brevet.

L'ISO (www.iso.org/patents) et l'IEC (<http://patents.iec.ch>) gèrent des bases de données en ligne de brevets relatifs à leurs normes. Les utilisateurs sont encouragés à consulter les bases de données pour disposer des informations les plus récentes concernant les brevets.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 4-25: Spécification du protocole de la couche liaison de données – Eléments de type 25

1 Domaine d'application

1.1 Généralités

La couche de liaison de données assure les communications de messagerie prioritaires de base entre les appareils d'un environnement d'automatisation.

Ce protocole offre des opportunités de communication à toutes les entités de liaison de données participantes

- a) des opportunités de communication cyclique à démarrage synchrone, selon un ordre préétabli, et
- b) de manière cyclique ou acyclique asynchrone, comme demandé par chaque cycle de chacune de ces entités de liaison de données.

Par conséquent, ce protocole peut se caractériser comme assurant un accès cyclique et acyclique asynchrone, mais avec un redémarrage synchrone de chaque cycle.

1.2 Spécifications

Le présent document spécifie

- a) les procédures de transfert en temps opportun des données et des informations de commande entre une entité utilisateur de liaison de données et une entité utilisateur homologue et, parmi les entités de liaison de données formant le fournisseur de services distribués de liaison de données;
- b) les procédures pour donner des opportunités de communication à toutes les entités DL participantes, séquentiellement et de manière cyclique pour le transfert déterministe et synchronisé à intervalles cycliques pouvant aller jusqu'à 64 ms;
- c) les procédures pour donner les opportunités de communication disponibles pour la transmission de données prioritaires ainsi que pour la transmission de données non prioritaires, sans nuire à la transmission de données prioritaire;
- d) les procédures pour donner des opportunités de communication cyclique et acyclique pour la transmission de données prioritaires avec accès hiérarchisés;
- e) les procédures pour donner des opportunités de communication selon le contrôle d'accès au support physique normalisé par l'ISO/IEC/IEEE 8802-3, avec des dispositions pour les nœuds à ajouter ou à retirer lors du fonctionnement normal;
- f) la structure des DLPDU de bus de terrain utilisées par le protocole du présent document pour le transfert des données et des informations de commande, et leur représentation sous forme d'unités de données d'interface physique.

1.3 Procédures

Les procédures sont définies en termes

- a) d'interactions entre les entités DL (DLE) homologues par l'échange de DLPDU de bus de terrain;

- b) d'interactions entre un fournisseur de service DL (DLS) et un utilisateur DLS au sein du même système par l'échange de primitives DLS;
- c) d'interactions entre un fournisseur DLS et un fournisseur de service Ph au sein du même système par l'échange de primitives de service Ph.

1.4 Applicabilité

Ces procédures s'appliquent aux instances de communication entre des systèmes qui prennent en charge des services de communications prioritaires dans la couche liaison de données des modèles de référence OSI ou de bus de terrain, et qui peuvent être connectés dans un environnement d'interconnexion de systèmes ouverts.

Les profils sont un moyen simple à plusieurs attributs de récapituler les capacités d'une mise en œuvre, et donc son applicabilité en fonction des différents besoins de communications prioritaires.

1.5 Conformité

Le présent document spécifie également les exigences de conformité relatives aux systèmes mettant en œuvre ces procédures. Le présent document ne comporte aucun essai visant à démontrer la conformité à ces exigences.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences 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).

NOTE Toutes les parties de la série IEC 61158, ainsi que l'IEC 61784-1 et l'IEC 61784-2 font l'objet d'une maintenance simultanée. Les références croisées à ces documents dans le texte se rapportent par conséquent aux éditions datées dans la présente liste de références normatives.

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

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

ISO/IEC/IEEE 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Standard for Ethernet* (disponible en anglais seulement)

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

IEEE Std 802.1D, *IEEE Standard for Local and metropolitan area networks – Media access Control (MAC) Bridges*, disponible sur <http://www.ieee.org> [consulté le 2018-09-17]

IEEE 802.1Q, *IEEE standard for Local and metropolitan area networks – Virtual bridged local area networks Bridges*; disponible à l'adresse <http://www.ieee.org> [consulté le 2018-09-17]

3 Termes, définitions, symboles, abréviations et conventions

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

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <http://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <http://www.iso.org/obp>

3.1 Termes et définitions du modèle de référence

Le présent document est basé en partie sur les concepts développés dans l'ISO/IEC 74981 et l'ISO/IEC 7498-3 et utilise les termes suivants définis:

| | | |
|---------------|--|------------------|
| 3.1.1 | adresse DL | [ISO/IEC 7498-3] |
| 3.1.2 | mapping d'adresses de DL | [ISO/IEC 7498-1] |
| 3.1.3 | adresse DL appelée | [ISO/IEC 7498-3] |
| 3.1.4 | adresse DL appelante | [ISO/IEC 7498-3] |
| 3.1.5 | connexion centralisée à points d'extrémité multiples | [ISO/IEC 7498-1] |
| 3.1.6 | connexion DL | [ISO/IEC 7498-1] |
| 3.1.7 | point d'extrémité de connexion DL | [ISO/IEC 7498-1] |
| 3.1.8 | identificateur de point d'extrémité de connexion de DL | [ISO/IEC 7498-1] |
| 3.1.9 | transmission en mode connexion DL | [ISO/IEC 7498-1] |
| 3.1.10 | transmission en mode sans connexion DL | [ISO/IEC 7498-1] |
| 3.1.11 | entités (N) correspondantes entités DL correspondantes (N=2) entités Ph correspondantes (N=1) | [ISO/IEC 7498-1] |
| 3.1.12 | transmission duplex de DL | [ISO/IEC 7498-1] |
| 3.1.13 | entité (N) [ISO/IEC 7498-1] entité DL (N=2) entité Ph (N=1) | |
| 3.1.14 | fonctionnalité de DL | [ISO/IEC 7498-1] |
| 3.1.15 | contrôle de flux | [ISO/IEC 7498-1] |
| 3.1.16 | couche (N) couche DL (N=2) couche Ph (N=1) | [ISO/IEC 7498-1] |
| 3.1.17 | gestion de couche | [ISO/IEC 7498-1] |
| 3.1.18 | vue locale de DL | [ISO/IEC 7498-3] |
| 3.1.19 | nom de DL | [ISO/IEC 7498-3] |
| 3.1.20 | domaine de dénomination-(adressage) | [ISO/IEC 7498-3] |
| 3.1.21 | entités homologues | [ISO/IEC 7498-1] |
| 3.1.22 | nom de la primitive | [ISO/IEC 7498-3] |
| 3.1.23 | protocole de DL | [ISO/IEC 7498-1] |
| 3.1.24 | identificateur de connexion de protocole de DL | [ISO/IEC 7498-1] |
| 3.1.25 | unité de données de protocole de DL | [ISO/IEC 7498-1] |
| 3.1.26 | relais de DL | [ISO/IEC 7498-1] |
| 3.1.27 | réinitialisation | [ISO/IEC 7498-1] |
| 3.1.28 | adresse DL répondante | [ISO/IEC 7498-3] |
| 3.1.29 | routage | [ISO/IEC 7498-1] |
| 3.1.30 | segmentation | [ISO/IEC 7498-1] |

| | | |
|---------------|---|------------------|
| 3.1.31 | couche (N) service DL (N=2) service Ph (N=1) | [ISO/IEC 7498-1] |
| 3.1.32 | point d'accès au service (N) point d'accès au service DL (N=2) point d'accès au service Ph (N=1) | [ISO/IEC 7498-1] |
| 3.1.33 | adresse de point d'accès au service DL | [ISO/IEC 7498-3] |
| 3.1.34 | identificateur de connexion de service DL | [ISO/IEC 7498-1] |
| 3.1.35 | unité de données de service DL | [ISO/IEC 7498-1] |
| 3.1.36 | transmission simplex DL | [ISO/IEC 7498-1] |
| 3.1.37 | sous-système de DL | [ISO/IEC 7498-1] |
| 3.1.38 | gestion de systèmes | [ISO/IEC 7498-1] |
| 3.1.39 | données d'utilisateur DLS | [ISO/IEC 7498-1] |

3.2 Termes et définitions de convention de service

Le présent document utilise également les termes suivants définis dans l'ISO/IEC 10731 tels qu'ils s'appliquent à la couche liaison de données:

| | |
|---------------|--|
| 3.2.1 | accepteur |
| 3.2.2 | service asymétrique |
| 3.2.3 | (primitive) "confirm"; (primitive) "requestor.deliver" |
| 3.2.4 | deliver (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 | fonctionnalité initiée de fournisseur DL |
| 3.2.11 | fonctionnalité facultative de fournisseur DL |
| 3.2.12 | primitive de service DL; primitive |
| 3.2.13 | fournisseur de service DL |
| 3.2.14 | utilisateur de service DL |
| 3.2.15 | fonctionnalité facultative d'utilisateur DLS |
| 3.2.16 | (primitive) "indication"; (primitive) "acceptor.deliver" |
| 3.2.17 | à plusieurs homologues |
| 3.2.18 | (primitive) "request"; (primitive) "requestor.submit" |
| 3.2.19 | demandeur |
| 3.2.20 | réponse (primitive); (primitive) "acceptor.submit" |
| 3.2.21 | submit (primitive) |
| 3.2.22 | service symétrique |

3.3 Termes et définitions

3.3.1

blocage

état du port qui ne prend pas part à la communication de trame

3.3.2

classe

identificateurs qui désignent la plage de communication de la trame RCL et des autres trames

3.3.3

communication de commande

communication de données acycliques en temps différé pour les applications à priorité plus élevée et communication de commande de nœud

3.3.4

transmission cyclique

communication de données périodique pour la communication en temps réel

3.3.5

adresse DLCEP

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

DLSAP (Point d'accès au service DL)

point distinctif au niveau duquel des DL-services sont fournis par une seule DL-entity à une seule entité de couche supérieure

3.3.7

DL(SAP)-address

soit une adresse DLSAP individuelle qui désigne un seul DLSAP d'un seul utilisateur DLS, soit un groupe d'adresses de DLS qui désigne potentiellement plusieurs DLSAP, chacun pour un seul utilisateur DLS

Note 1 à l'article: Cette terminologie est choisie parce que l'ISO/IEC 7498-3 n'autorise pas l'utilisation du terme adresse DLSAP pour désigner plus d'un DLSAP pour un seul utilisateur DLS.

3.3.8

adresse DLSAP (individuelle)

DL-address qui désigne un seul DLSAP dans la liaison étendue

Note 1 à l'article: Une seule DL-entity peut comporter plusieurs DLSAP-addresses associées à un seul DLSAP.

3.3.9

liaison étendue

sous-réseau DL, constitué d'un ensemble maximal de liaisons interconnectées par des relais de DL, qui partagent un seul espace de nom de DL (adresse DL) dans lequel n'importe quelle entité de DL connectée peut communiquer avec l'autre, soit directement, soit au moyen d'une ou de plusieurs entités de relais de DL intervenantes

3.3.10

trame

synonyme déconseillé de DLPDU

3.3.11

Adresse DL de groupe

adresse DL qui désigne potentiellement plus d'un DLSAP au sein d'une liaison étendue

Note 1 à l'article: Une seule entité de DL peut comporter plusieurs adresses de DL de groupe associées à un seul DLSAP. Une seule entité de DL peut également posséder une seule adresse DL de groupe associée à plusieurs DLSAP.

3.3.12

communication d'information

communication de données acycliques en temps différé pour les applications non prioritaires

3.3.13

liaison logique déconnectée

statut de liaison dans lequel le port est dans un état de blocage et ne communique pas tous les types de trame, à l'exception des trames RCL

3.3.14

liaison logique établie

statut de liaison dans lequel le port communique tous les types de trame

3.3.15

nœud

DL-entity unique telle qu'elle se présente sur une liaison locale

3.3.16

liaison physique déconnectée

statut de liaison dans lequel le port ne communique pas les trames à cause du statut de déconnexion de liaison défini dans l'ISO/IEC/IEEE 8802-3

3.3.17

utilisateur DLS destinataire

utilisateur de service DL auquel sont destinées les données utilisateur DL

Note 1 à l'article: Un utilisateur de DL-service peut être simultanément un sending DLS-user et un receiving DLS-user.

3.3.18

communication de commande d'anneau (RCL)

communication de commande du réseau d'anneau DLL de Type 25 utilisant des trames RCL et des trames différées

3.3.19

utilisateur DLS expéditeur

utilisateur du service DL à la source des données utilisateur DL

3.3.20

poste

synonyme de nœud

3.3.21

adresse station

adresse d'identificateur qui désigne le nœud du réseau de Type 25

3.4 Symboles et abréviations

NOTE La plupart des symboles et abréviations sont communs à plusieurs types de protocole et ne sont pas nécessairement utilisés par chacun d'eux.

| | |
|--------|---|
| DL- | Data-link layer (as a prefix (Préfixe désignant la couche liaison de données) |
| DLC | DL-connection (connexion DL) |
| DLCEP | DL-connection-end-point (point d'extrémité de connexion DL) |
| DLE | DL-entity (Entité DL) (instance active locale de la couche liaison de données) |
| DLL | DL-layer (Couche liaison de données) |
| DLM | DL-management (Gestion DL) |
| DLME | DL-management entity (Entité de gestion de DL) (instance active locale de la gestion de DL) |
| DLMS | DL-management service (Service de gestion de DL) |
| DLPCI | DL-protocol-control-information (informations de commande de protocole de DL) |
| DLPDU | DL-protocol-data-unit (unité de données de protocole DL) |
| DLS | DL-service (Service DL) |
| DL SAP | DL-service-access-point (Point d'accès au service DL) |
| DLSDU | DL-service-data-unit (Unité de données de service de liaison de données) |
| EGA | Edge-A node (Nœud Edge-A) |
| EGB | Edge-B node (Nœud Edge-B) |
| FIFO | First-in First-out (Premier entré, premier sorti) (méthode de mise en file d'attente) |
| ISL | Isolate node (Nœud isolant) |
| ITM | Intermediate node (Nœud intermédiaire) |
| LCA | Loop condition alert (Alarme d'état de boucle) (type de trame de Type 25) |
| LCC | Loop condition check (Contrôle de l'état de la boucle) (type de trame de Type 25) |
| LCN | Loop condition notify (Notification de l'état de boucle) (type de trame de Type 25) |
| LLD | Logical link down (Liaison logique déconnectée) |
| LLU | Logical link up (Liaison logique établie) |
| LNA | Loop notify answer (Réponse à un avertissement de boucle) (type de trame de Type 25) |
| NNB | No neighborhood state (Pas d'état de voisinage) |
| OSI | Open systems interconnection (Interconnexion de systèmes ouverts) |
| Ph- | Physical layer (Préfixe désignant la couche physique) |
| PhE | Ph-entity (entité Ph) (instance active locale de la couche physique) |
| PhL | Ph-layer (Couche physique) |
| PhS | Ph-service (service Ph) |
| PLD | Physical link down (Liaison physique déconnectée) |
| PLU | Port A (B) link up state (État de liaison établie du port A (B)) |
| QoS | Qualité de service |
| RCL | Quality-of-service (Commande d'anneau) |
| RCLC | RCL communication control (commande de communication RCL) |

| | |
|-----|---|
| RHE | Rapid hello (Hello rapide) (type de trame de Type 25) |
| RT | Real time (temps réel) |
| RTC | RT communication control (Contrôle de communication RT) |
| SCR | Station condition report (Rapport d'état de poste) (type de trame de Type 25) |
| TRC | Transmit/Receive control (Commande de transmission/réception) |
| WLU | Wait link up state (État d'attente de liaison active) |

3.5 Conventions communes

Le présent document utilise les conventions descriptives données dans l'ISO/IEC 10731.

Le modèle de service, les primitives de service et les schémas de séquence de temps utilisés sont des descriptions entièrement abstraites; elles ne correspondent pas à une spécification de mise en œuvre.

Les primitives de service, utilisées pour représenter les interactions entre utilisateur de service et fournisseur de service (voir ISO/IEC 10731), véhiculent les paramètres qui désignent les informations disponibles dans l'interaction utilisateur/fournisseur.

Le présent document utilise des tableaux pour décrire les paramètres de composants des primitives DLS. Les paramètres qui s'appliquent à chaque groupe de primitives DLS sont définis dans des tableaux jusqu'à la fin du présent document. Chaque tableau comporte jusqu'à six colonnes, contenant le nom du paramètre de service, avec une colonne pour ces primitives et pour les directions de transfert de paramètres utilisés par le DLS:

- les paramètres d'entrée de la primitive "request" (demande);
- les paramètres de sortie de la primitive "request" (demande);
- les paramètres de sortie de la primitive "indication";
- les paramètres d'entrée de la primitive "response" (réponse);
- les paramètres de sortie de la primitive "confirm" (confirmation).

NOTE Les primitives de demande, indication, réponse et confirmation sont également connues comme primitives requestor.submit (demandeur.soumission), acceptor.deliver (accepteur.remise), acceptor.submit (accepteur.soumission) et requestor.deliver (demandeur.remise), respectivement (voir ISO/IEC 10731).

Un paramètre (ou un composant de celui-ci) est énuméré dans chaque ligne de chaque tableau. Dans les colonnes appropriées de la primitive de service, un code est utilisé pour spécifier le type d'usage du paramètre sur la primitive spécifiée dans la colonne:

- M** – le paramètre est obligatoire pour la primitive.
- U** – Le paramètre est une option de l'utilisateur et peut ou peut ne pas être fourni, cela dépendant de l'usage dynamique de l'utilisateur DLS. Lorsqu'il n'est pas fourni, une valeur par défaut est supposée pour le paramètre;
- C** – Le paramètre est conditionné à d'autres paramètres ou à l'environnement de l'utilisateur DLS.

(blanc/vide) – Le paramètre n'est jamais présent.

Certaines entrées sont également qualifiées par des éléments entre parenthèses. Il peut s'agir

a) une contrainte spécifique au paramètre:

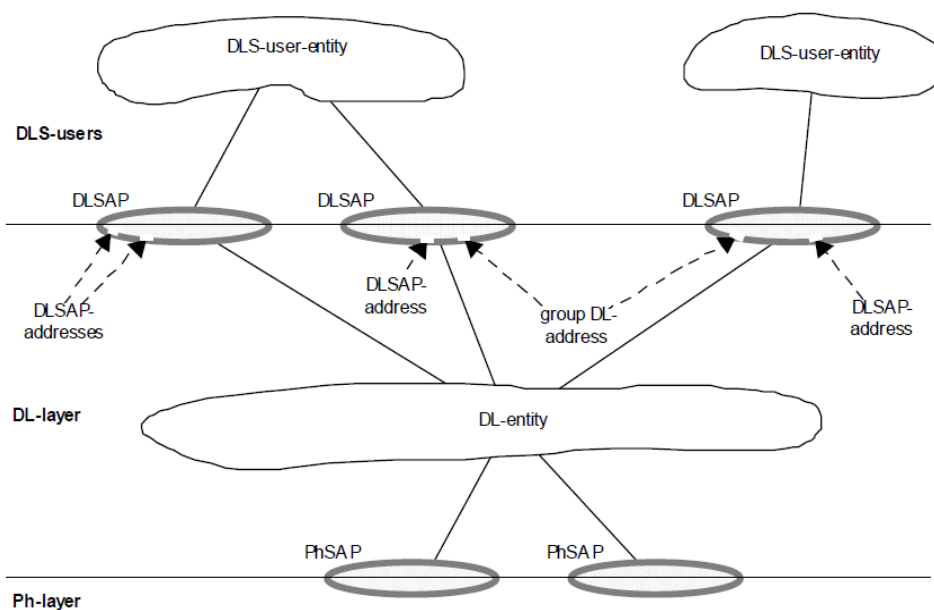
- (=)** "(=)" indique que le paramètre équivaut sur le plan sémantique au paramètre de la primitive de service se trouvant immédiatement à sa gauche dans le tableau.

b) une indication qu'une certaine note s'applique à l'article:

- n)** (n) indique que la note suivante n contient des informations supplémentaires relatives au paramètre et à son utilisation.

Dans une interface particulière, il n'est pas nécessaire de déclarer explicitement tous les paramètres. Certains peuvent être associés de façon implicite au DLSAP en lequel la primitive est émise.

La Figure 1 montre les Relations des DLSAP, des adresses de DLSAP et des adresses de DL de groupe.



| Anglais | Français |
|------------------|-------------------------------------|
| DLS-user-entity | Entité utilisateur DLS |
| DLS-users | Utilisateurs de DLS |
| DLSAP | DLSAP (Point d'accès au service DL) |
| DLSAP-addresses | Adresses DLSAP |
| Group DL-address | Adresse DL de groupe |
| DL-layer | Couche liaison de données |
| DL-entity | Entité de DL |
| PhSAP | PhSAP |
| Ph-layer | Couche physique |

NOTE 1 Des DLSAP et des PhSAP sont représentés par des ovals qui couvrent la frontière entre deux couches adjacentes.

NOTE 2 Des adresses de DL sont représentées par des petites lacunes (points d'accès) dans la portion de DLL d'un DLSAP.

NOTE 3 Une seule entité DL peut comporter plusieurs adresses DLSAP et adresses DL de groupe associées à un seul DLSAP.

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

Dans les diagrammes qui illustrent ces interfaces, des traits discontinus indiquent les relations cause-effet ou temps-séquence, alors que des lignes ondulées indiquent que les événements sont approximativement contemporains.

3.6 Conventions de type 25 supplémentaires

3.6.1 Conventions pour les primitives

La notation suivante, forme raccourcie des classes de primitive définies en 3.2, est utilisée dans les figures.

- req** primitive de demande
- ind** primitive d'indication
- cnf** primitive de confirmation
- rsp** response primitive (primitive "response")

3.6.2 Conventions dans les diagrammes d'états

Les séquences de protocole sont décrites au moyen de diagrammes d'états.

Dans les diagrammes d'état, les états sont représentés par des cases et les transitions d'état par des flèches.

Les noms des états et des transitions du diagramme d'état correspondent aux noms indiqués dans la table d'états. Cette liste de transitions d'états est structurée comme indiqué dans le Tableau 1.

Tableau 1 – Descriptions des transitions d'état

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|--------------|--------------------------------|--------------|
| | | | |

Tableau 2 – Description des éléments d'un diagramme d'états

| Élément de description | Signification |
|-------------------------------|---|
| # | Numéro de la transition. |
| État courant, état suivant | Noms de l'état de départ et de l'état de transition cible. |
| Événement | Nom ou description de l'événement déclencheur qui provoque la transition. |
| Conditions | Expression booléenne, qui doit être vraie pour que la transition ait lieu. |
| =>actions | Liste des affectations et des appels de service ou de fonction. Il convient que l'action soit atomique. Le symbole "=>" qui précède ne fait pas partie de l'action. |
| "/ conditions" peut être omis | |

Les conventions utilisées dans les diagrammes d'états sont présentées dans le Tableau 3.

Tableau 3 – Conventions utilisées dans les diagrammes d'états

| Convention | Signification |
|------------|---|
| = | Substitution du côté droit par le côté gauche |
| == | Condition logique indiquant qu'une entité de gauche est égale à une entité de droite. |
| != | Condition logique indiquant qu'une entité de gauche n'est pas égale à une entité de droite. |
| < | Condition logique indiquant qu'une entité de gauche est inférieure à une entité de droite. |
| > | Condition logique indiquant qu'une entité de gauche est supérieure à une entité de droite. |
| && | Logique AND |
| | "OR" (c'est-à-dire: OU) logique |
| ! | Opérateur de négation |
| + - * / | Opérateur arithmétique |
| ++ | Incrémenter l'opérateur |
| -- | Décrémenter l'opérateur |
| ; | Point d'interruption |

4 Vue d'ensemble du DL-protocole

4.1 Généralités

Un bus de terrain de Type 25 est basé sur l'ISO/IEC/EEE 8802-3, mais est étendu vers l'Ethernet en temps réel. Le réseau de Type 25 contrôle le trafic de données et attribue des priorités de trame avec le VLAN IEEE 802.1Q.

NOTE Tout composant d'infrastructure Ethernet au silicium normalisé ou tout équipement de mesure et d'essai normalisé, tel qu'un analyseur de réseau, s'applique à un réseau de Type 25.

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

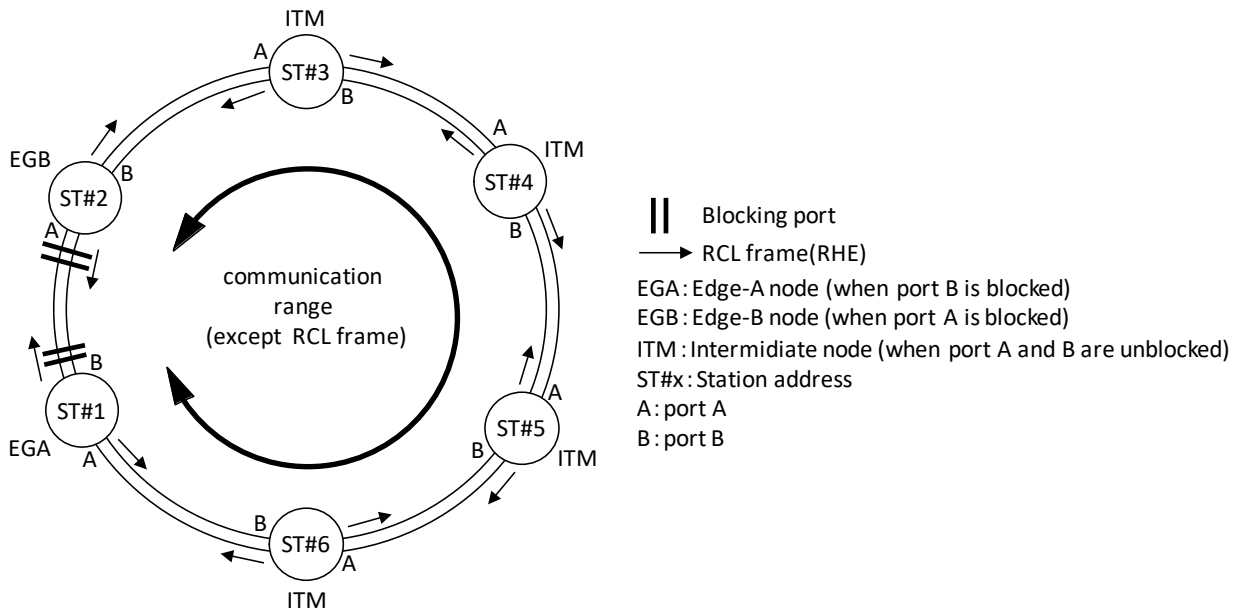
4.2.1 Généralités

Le réseau de Type 25 fournit les mécanismes permettant d'obtenir les résultats suivants.

- a) Il attribue des priorités à quatre types de trame (RCL/Cyclique/Commande/Information).
- b) Les trames du réseau de Type 25 sont garanties en temps réel de la communication cyclique pour commander le trafic de données.
- c) Elle utilise la communication ISO/IEC/IEEE 8802-3 sans avoir besoin de fonctions spéciales pour la communication RTE, telles que le partage temporel ou la synchronisation des nœuds.
- d) La topologie en anneau est uniquement admise dans le réseau de Type 25. Afin de commander le réseau, il classe les trames en trois classes en fonction de leurs plages de communication.

4.2.2 La topologie du réseau

La topologie en anneau doit être utilisée dans le réseau de Type 25. Le réseau doit être commandé et reconfiguré par des trames RCL. Le réseau en anneau de Type 25 ne nécessite pas de concept de modèle maître/esclave, car chaque nœud du réseau de Type 25 se reconfigure de manière autonome. Pour éviter la boucle de trame, une section du réseau en anneau est à l'état de blocage (voir l'IEEE 802.1D) et la section bloquée met fin à tous les types de trame, à l'exception de la trame RCL (comme illustré au Tableau 2).



| Anglais | Français |
|---|--|
| ITM | ITM |
| ST# | ST |
| communication range (except RCL frame) | plage de communication (sauf trame RCL) |
| Blocking port | Port bloqué |
| RCL frame(RHE) | trame RCL (RHE) |
| EGA: Edge-A node (when port B is blocked) | EGA: Nœud Edge-A (lorsque le port B est bloqué) |
| EGB: Edge-B node (when port A is blocked) | EGB: Nœud Edge-B (lorsque le port A est bloqué) |
| ITM : Intermediate node (when port A and B are unblocked) | ITM: Nœud d'interconnexion (lorsque les ports A et B sont débloqués) |
| ST#x : Station address | ST#x: Adresse station |
| A: port A | A: port A |
| B: port B | B: port B |

Figure 2 – Commande d'anneau dans le réseau de Type 25

Chaque nœud du réseau en anneau comporte deux ports A et B pour la connexion aux nœuds voisins. Chaque nœud a sa propre adresse de poste. Dans le réseau de Type 25, chaque nœud a quatre états de nœud différents en fonction de statut de blocage de leurs ports. Le Tableau 4 présente les caractéristiques des états du nœud.

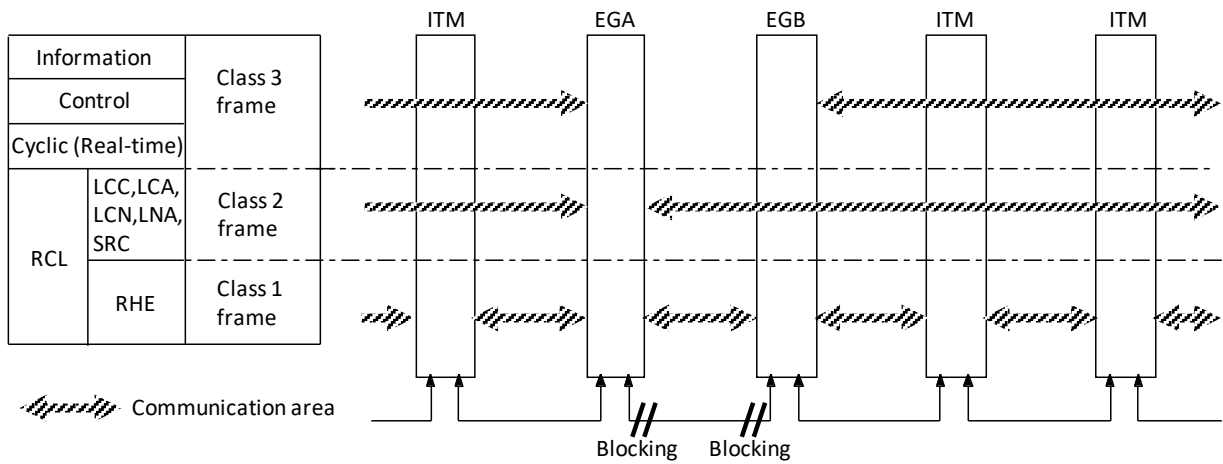
Tableau 4 – Caractéristiques des états du nœud

| Etat du nœud | Abréviation | Description |
|--|-------------|--|
| Nœud isolé | ISL | Les ports A et B sont tous deux bloqués dans cet état (par exemple, pas de connexion de câbles à d'autres nœuds) et le nœud est isolé du réseau de Type 25. |
| Nœud Edge-A (lorsque le port B est bloqué) | EGA | Le port B est bloqué, le port A n'est pas bloqué. Le nœud fonctionne avec Edge-A. Il n'y a qu'un seul nœud dans un réseau de Type 25. |
| Nœud Edge-B (lorsque le port A est bloqué) | EGB | Le port A est bloqué, le port B n'est pas bloqué. Le nœud fonctionne avec Edge-B. Il n'y a qu'un seul nœud dans un réseau de Type 25. |
| Nœud intermédiaire (lorsque les port A, B sont débloqués) | ITM | Les ports A et B sont tous deux débloqués. Le nœud fonctionne avec Edge-A et Edge-B. Dans un réseau de Type 25, il y a un nœud Edge-A et un nœud Edge-B. Les autres nœuds sont cet état. |

Les trames RCL sont basées sur les trames générales ISO/IEC/IEEE 8802-3. Ces trames ont des plages de communication différentes, qui sont données par la « classe ». Le Tableau 5 montre les caractéristiques de la classe de trames et Figure 3 les plages de communication dans le réseau de Type 25.

Tableau 5 – Caractéristique des classes de trame

| Classe de trame | Description |
|-----------------|--|
| Classe 1 | Cette trame est utilisée pour communiquer avec le nœud voisin uniquement. Elle traverse les ports de blocage. Une trame RHE correspond à des trames RCL dans cette classe. |
| Classe 2 | Cette trame sert à communiquer autour du réseau. Cette trame est capable de traverser les ports de blocage et le nœud Edge-A met fin aux trames. Les trames LCC, LCA, LCN, LNA et SCR sont des trames RCL de cette classe. |
| Classe 3 | Cette trame est une trame Ethernet générale et est arrêtée par des ports de blocage. Les trames RT (trames cycliques, de commande, d'information) sont dans cette classe. |



| Anglais | Français |
|-------------------------|-------------------------|
| Information | Informations |
| Control | Commande |
| Cyclic (Real-time) | Cyclique (temps réel) |
| RCL | RCL |
| LCC, LCA, LCN, LNA, SRC | LCC, LCA, LCN, LNA, SRC |
| RHE | RHE |
| Class 3 frame | Trame de classe 3 |
| Class 2 frame | Trame de classe 2 |
| Class 1 frame | Trame de classe 1 |
| ITM | ITM |
| EGA | EGA |
| EGB | EGB |
| Communication area | Zone de communication |
| Blocking | Blocage |

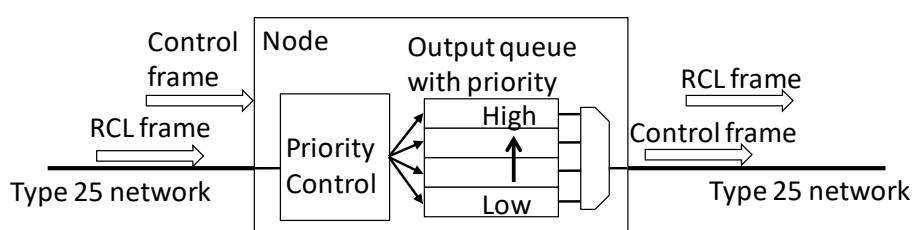
Figure 3 – Plages de communication des trames de Type 25

4.2.3 Contrôle de priorité avec VLAN

Le Tableau 6 montre le mapping de priorité VLAN du réseau de Type 25. Le réseau de Type 25 mappe la priorité VLAN sur quatre types de communication: commande d'anneau, cyclique, commande, information. Communication de commande d'anneau: la trame de reconfiguration de réseau en anneau de Type 25 est la priorité la plus élevée. Communication cyclique: la trame de communication temps réel est la deuxième priorité la plus élevée. La communication de commande est la troisième priorité et l'information est la priorité la plus basse.

Tableau 6 – Mapping de priorité VLAN du réseau de Type 25

| VLAN | Utilisation | Priorité VLAN |
|---------|------------------------------------|---------------------------|
| VLAN-1. | Communication de commande d'anneau | 7 (valeur la plus élevée) |
| VLAN-2. | transmission cyclique | 5 (haut) |
| VLAN-3. | Communication de commande | 3 (bas) |
| VLAN-4. | Communication d'information | 1 (valeur la plus faible) |



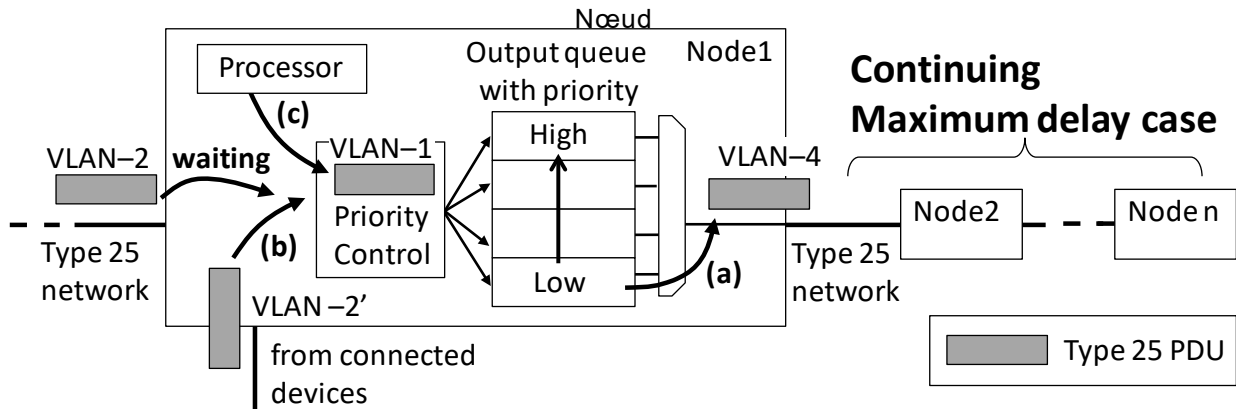
| Anglais | Français |
|----------------------------|--|
| Control frame | Trame de commande |
| RCL frame | Trame RCL |
| Type 25 network | Réseau de Type 25 |
| Node | Nœud |
| Priority Control | Contrôle de priorité |
| Output queue with priority | File d'attente de sortie avec priorité |
| High | Température |
| Low | Basse |

Figure 4 – Contrôle de priorité avec VLAN de réseau de Type 25

Chaque nœud a des files d'attente de sortie avec priorité. Les trames de réception sont placées dans les files d'attente de sortie en fonction de leurs priorités de trame. La trame de priorité plus élevée sera transférée vers le nœud voisin. Dans ce mécanisme, le réseau de Type 25 garantit la communication cyclique en temps réel.

4.2.4 Retard de remise maximal dans le réseau de Type 25

Le réseau de Type 25 limite le trafic de données dans le réseau pour assurer la communication en temps réel. Ceci maintient le retard de communication des trames à un niveau acceptable. La restriction de trafic de chaque nœud est déterminée par le nombre total de nœuds du réseau et la durée de cycle utilisant le nœud.



| Anglais | Français |
|-------------------------------|---|
| VLAN-2 | VLAN-2. |
| Type 25 network | Réseau de Type 25 |
| Processor | Processeur |
| waiting | attente |
| VLAN-1 | VLAN-1. |
| Priority Control | Contrôle de priorité |
| From connected devices | En provenance des dispositifs connectés |
| Output queue with priority | File d'attente de sortie avec priorité |
| High | Température |
| Low | Basse |
| Node1 | Node1. |
| Continuing Maximum delay case | Cas de retard maximal continu |
| Node2 | Node2. |
| Noden | Node n |
| Type 25 PDU | PDU Type 25 |

Figure 5 – Mécanisme de retard de transmission d'un nœud

Il existe trois causes de retard concernant la trame VLAN-2 (communication cyclique) dans le réseau de Type 25.

- a) La trame VLAN-2 arrive au moment où le nœud envoie une autre trame. La trame VLAN-2 est forcée d'attendre jusqu'à ce que la transmission soit terminée. Ce retard est causé par des trames prioritaires.
- b) Avant de recevoir une trame VLAN-2, le nœud reçoit une autre trame VLAN-2 (VLAN-2' dans la Figure 5) en provenance d'autres ports. Dans ce cas, au niveau de la file d'attente de priorité, la trame VLAN2 est placée après la trame VLAN-2'.
- c) Lorsque la trame VLAN-2 attend une transmission pour les raisons a) et b), le nœud reçoit des trames de priorité plus élevée (trame VLAN-1). Comme le nœud transmet d'abord les trames ayant la priorité la plus élevée, la trame VLAN-2 est forcée d'attendre que la transmission des trames de priorité plus élevée soit terminée.

Lorsque trois causes se produisent en même temps dans un nœud, le retard maximal se produit. Le temps de remise peut être calculé en cas de retard maximal survenant au niveau de tous les nœuds d'un réseau de Type 25.

4.2.5 Commande de trafic pour la communication en temps réel

Le temps de remise dans le réseau de Type 25 est calculé en fonction du temps de retard maximal par nœud, du nombre de nœuds et de la longueur du réseau en anneau.

Le retard maximal de communication en temps réel (communication cyclique) entre deux nœuds est calculé en tenant compte des quatre causes de retard présentées à la Figure 6.

1) Délai d'empilage au niveau du nœud émetteur:

Le retard causé par le processus de transmission. Il dépend de la mise en œuvre du matériel et du logiciel.

2) Retard de segment de câble:

Le retard causé par le réseau en anneau de construction de câble de transmission. Il est proportionnel à la longueur du câble.

3) Retard de transmission dans le nœud de transit

Le retard dû au retard de transmission par nœud de transit. Il est proportionnel au nombre de nœuds du chemin de transmission. Le détail est présenté en 4.2.4.

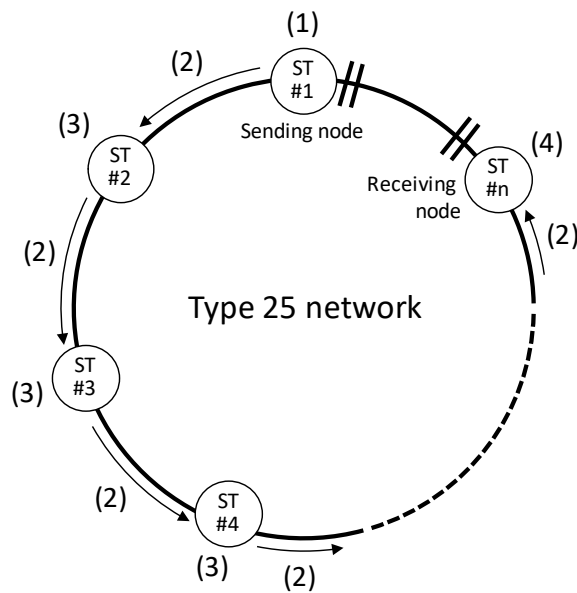
4) Retard d'empilage dans le nœud récepteur:

Le retard causé par le processus de réception. Il dépend de la mise en œuvre du matériel et du logiciel.

Le temps de remise est calculé par la formule suivante.

$$(\text{temps de remise}) = (1) + (2) \times \text{longueur de câble} + (3) \times (\text{nombre de nœuds} - 2) + (4)$$

La durée de cycle de la communication cyclique doit être fixée à une valeur supérieure au temps de remise, afin de garantir la communication en temps réel des trames de communication cyclique dans le réseau de Type 25.



| Anglais | Français |
|-----------------|-------------------|
| Sending node | Nœud émetteur |
| Receiving node | Nœud récepteur |
| Type 25 network | Réseau de Type 25 |
| ST# | ST |

Figure 6 – Retard maximal du réseau de Type 25

4.3 Service pris en charge à partir de la PhL

Le paragraphe 4.3 décrit le service physique (PhS) pris en charge et les contraintes utilisées par la DLE. Le service physique est supposé fournir les primitives de service suivantes, spécifiées dans l'Article 6 de l'ISO/IEC/IEEE 8802-3:2017.

Les primitives de PhS prises en charge sont:

- PHS-A_DATA.req;
- PHS-A_DATA.ind;
- PHS-B_DATA.req;
- PHS-B_DATA.ind.

4.4 Architecture de la couche DL

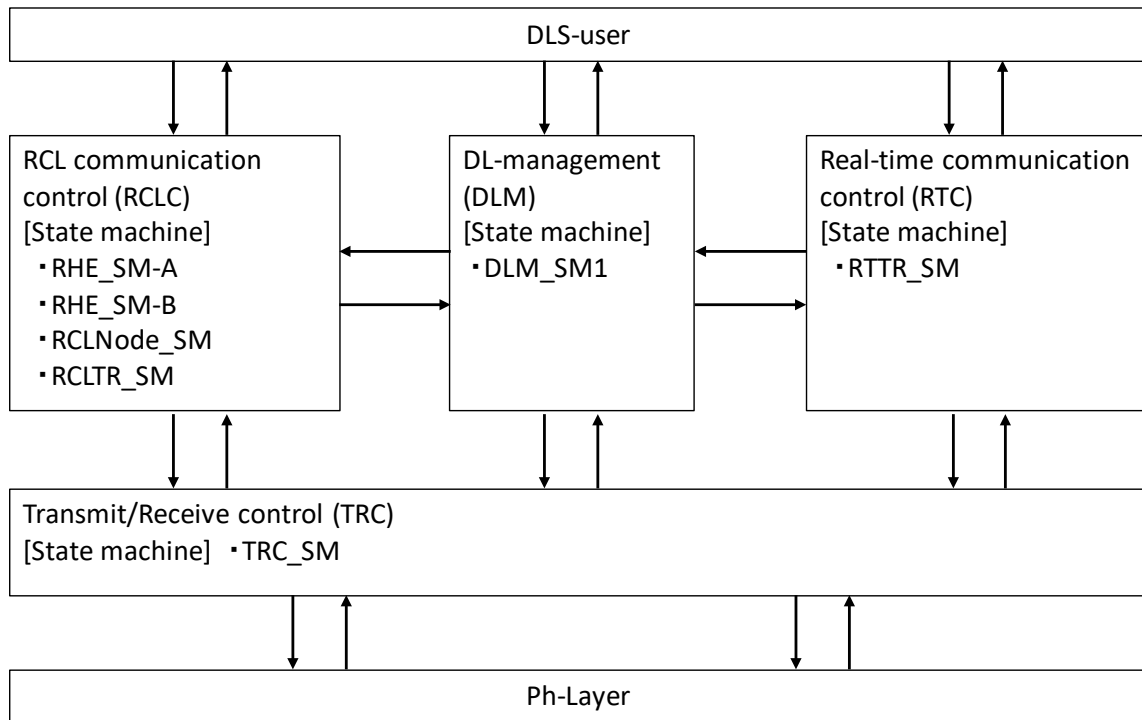
La DLL de bus de terrain de Type 25 est modélisée comme une combinaison de composants de commande de la commande de communication RCL (RCLC), de la commande de communication temps réel (RTC), de la commande de transmission/réception (TRC) et de la gestion DL (DLM).

La couche de liaison de données contient les composants figurant dans le Tableau 7.

Tableau 7 – Composants de la couche de liaison de données

| Composants | Description |
|---|---|
| Commande de communication RCL (RCLC) | Au niveau de la RCLC, la DLSDU pour les données de contrôle d'anneau (RCL) est stockée et transférée entre l'utilisateur DLS et la TRC. |
| Commande de communication en temps réel (RTC) | Au niveau de la RTC, les données de commande/d'information sont stockées et transférées entre l'utilisateur DLS et la TRC. |
| Commande de transmission/réception (TRC) | Au niveau de la TRC, la DLSDU est stockée et transférée de la RCLC/RTC vers la couche Ph. La DLPDU est stockée et transférée de la couche Ph vers la RCLC/RTC. |
| Gestion DL (DLM) | L'interface de gestion de DL détient des variables de gestion de poste 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 7. Les têtes de flèche représentent la direction principale du flux de données et du contrôle.



| Anglais | Français |
|---------------------------------------|---|
| DLS-user | Utilisateur DLS |
| RCL communication control (RCLC) | Commande de communication RCL (RCLC) |
| [State machine] | Diagramme d'états |
| RHE_SM-A | RHE_SM-A |
| RHE_SM-B | RHE_SM-B |
| RCLNode_SM | RCLNode_SM |
| RCLTR_SM | RCLTR_SM |
| DM-management (DLM) | Gestion DL (DLM) |
| [State machine] | Diagramme d'états |
| DLM_SM1 | DLM_SM1. |
| Real-time communication control (RTC) | Commande de communication en temps réel (RTC) |
| RTTR_SM | RTTR_SM |
| Transmit/receive control (TRC) | Commande de transmission/réception (TRC) |
| [State machine] | Diagramme d'états |
| TRC_SM | TRC_SM |
| Ph-Layer | Couche physique |

Figure 7 – Architecture interne de la couche de liaison de données

4.5 Paramètres et variables locaux

4.5.1 Vue d'ensemble

La présente spécification utilise les paramètres de demande d'utilisateur DLS P(...) et les variables locales V(...) comme moyen pour clarifier l'effet de certaines actions et les conditions sous lesquelles ces actions sont valides, les temporisateurs locaux T(...) comme moyen de surveiller les actions du fournisseur de DLS distribué et de garantir une réponse DLE locale en cas d'absence de ces actions, et les compteurs locaux C(...) pour exécuter des fonctions de mesure de vitesse.

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 autorisée si aucune valeur par défaut n'est spécifiée;
- b) Tous les compteurs doivent être initialisés à zéro;
- c) tous les temporisateurs doivent être initialisés à l'état inactif;

La gestion de DL peut modifier les valeurs des variables de configuration.

4.5.2 Variables, paramètres, compteur et temporisateur

4.5.2.1 P(RHE_RxCK_NUM)

Ce paramètre indique le temps d'expiration suite à la non-réception des trames RHE.

4.5.2.2 P(RHE_LKUP_NUM)

Ce paramètre indique le nombre de trames RHE de réception pour la liaison logique.

4.5.2.3 P(LCC_STOP_NUM)

Ce paramètre indique le numéro de seuil pour arrêter la transmission de trame LCC.

4.5.2.4 C(PA_RHE_LKUP), C(PB_RHE_LKUP)

Ces compteurs indiquent le nombre de trames RHE de réception continue au niveau du port A et du port B, respectivement.

4.5.2.5 C(PA_RHE_RxCK), C(PB_RHE_RxCK)

Ces compteurs sont utilisés pour vérifier les trames RHE reçues continuellement ou supprimées. Chaque valeur est diminuée dans T(PA_RHE_TIME) ou T(PB_RHE_TIME).

4.5.2.6 C(PA_Rx-RHE_NotMatch), C(PB_Rx-RHE_NotMatch)

Ces compteurs sont utilisés pour détecter le changement du nœud de voisinage pour un autre nœud.

4.5.2.7 C(Rx_Counter)

Ce compteur indique le nombre de trames LCC de réception par son propre nœud.

4.5.2.8 T(PA_RHE_TIME), T(PB_RHE_TIME)

T(PA_RHE_TIME) et T(PB_RHE_TIME) sont utilisés pour le cycle fixe afin d'effectuer le fonctionnement du RHE.

4.5.2.9 V(PA_Blocking), V(PB_Blocking)

Ces variables contiennent et désignent respectivement le statut de blocage sur les ports. Lorsque le port est bloqué, ces variables sont mises à « True », sinon à « False ».

4.5.2.10 V(LCN_Pri)

Cette variable est une valeur unique utilisée pour décider d'un nœud Edge-A dans le réseau. Cette variable est utilisée lorsque le nœud envoie des trames LCN.

4.5.2.11 V(LNA_Pri)

Cette variable est une valeur unique utilisée pour décider d'un nœud Edge-A dans le réseau. Cette variable est utilisée lorsque le nœud envoie des trames LNA.

4.5.2.12 V(PA_NBST_LKST), V(PB_NBST_LKST)

Ces variables contiennent et désignent le statut de liaison d'un nœud de voisinage. Il prend trois types de valeurs (NNB, WLU et PLU). Ces valeurs sont mises à jour par la réception des trames RHE.

4.5.2.13 V(PA_NBST_NDST), V(PB_NBST_NDST)

Cette variable contient et désigne le statut de nœud d'un nœud de voisinage. Il prend quatre types de valeurs (ISL, EGA, EGB et ITM) et met à jour la valeur par la réception des trames RHE.

4.5.2.14 V(PA_NBST_RCLADD), V(PB_NBST_RCLADD)

Ces variables contiennent et désignent l'adresse d'un nœud de voisinage. Ces valeurs sont mises à jour par la réception des trames RHE.

4.5.2.15 V(PA_RHE_CYCLE), V(PB_RHE_CYCLE)

Ces variables contiennent et désignent la durée de cycle à effectuer pour le fonctionnement de RHE.

4.5.2.16 V(PA_RHE_RxSeq), V(PB_RHE_RxSeq)

Ces variables contiennent et désignent le numéro de séquence des trames RHE de réception. Si des trames RHE provenant du nœud de voisinage sont reçues en continu, ces valeurs augmentent.

4.5.2.17 V(PA_STATUS), V(PB_STATUS)

Ces variables contiennent et désignent le statut de liaison de port du propre nœud. Elles prennent trois valeurs (NNB, WLU et PLU).

5 Structure générale et codage des PhIDU et DLPDU, mais aussi éléments de procédure connexes

5.1 Vue d'ensemble

La DLL et ses modes opératoires sont nécessaires pour fournir les services proposés à l'utilisateur DLS à l'aide des services disponibles auprès de la PhL. L'Article 5 décrit la structure et la sémantique de la DLPDU et la procédure, communément utilisées dans la présente spécification.

5.2 Structure, codage et éléments de trame MAC commune

5.2.1 Structure de trame MAC

La Figure 8 montre la DLPDU de bus de terrain de Type 25.

| | | | | | | | | |
|-----------------|------------|----------------------------|-----------------------|-----------------|---------------------|---------------|---------|------------|
| Preamble (7) | SFD (1) | Destination address (6) | Source address (6) | VLAN tag (4) | Length/ Type (2) | DLS-user data | Padding | FCS (4) |
|-----------------|------------|----------------------------|-----------------------|-----------------|---------------------|---------------|---------|------------|

(*the number in parenthesis indicates length of the field in octets

| Anglais | Français |
|-------------------------|----------------------------|
| Preamble (7) | Préambule (7) |
| SFD (1) | SFD (1) |
| Destination address (6) | Adresse de destination (6) |
| Source address (6) | Adresse source (6) |
| VLAN tag (4) | Étiquette VLAN (4) |
| Length/Type (2) | Longueur/Type (2) |
| DLS-user data | Données d'utilisateur DLS |
| Padding | Remplissage |
| FCS (4) | FCS (4) |

Figure 8 – TFormat de la trame de DLPDU de bus de terrain de Type 25

5.2.2 Éléments de la trame MAC

5.2.2.1 Préambule

Le champ de préambule est identique à celui décrit dans l'Article 3 de l'ISO/ ISO/IEC 8802-3:2017. Le champ de préambule est un champ de 56 bits utilisé pour permettre aux circuits de la partie de signalisation physique d'atteindre leur synchronisation de régime établi avec la chronologie des trames de réception.

Le motif du préambule est:

10101010 10101010 10101010 10101010 10101010 10101010 10101010

5.2.2.2 Délimiteur de début de trame (SFD)

Le délimiteur de début de trame (SFD) est identique à celui décrit dans l'Article 3 de l'ISO/IEC 8802-3:2017. Le champ SFD est la séquence de motif binaire "10101011". Il suit immédiatement le motif du préambule et indique le début de la trame.

5.2.2.3 Adresse de destination

L'adresse de destination est identique à celle décrite dans l'Article 3 de l'ISO/ ISO/IEC 8802-3:2017. Ce champ d'adresse a une longueur de 48 bits. Les trames reçues et envoyées au niveau de chaque nœud sont classées en trois classes (classes 1, 2 et 3) selon la plage de communication. Le détail de la classe est indiqué en 4.2.2.

La classe de trame se distingue par son adresse de destination. Le Tableau 8 montre l'adresse de destination de trois types de classes.

Tableau 8 – Format de l'adresse de destination

| Classe de trame | Type de trame | Adresse de destination |
|-----------------|-------------------------------|--|
| Classe 1 | RCL (RHE) | 01-80-C2-00-00-0F. |
| Classe 2 | RCL (LCC/LCA/LCN/LNA/SCR) | 01-80-C2-00-00-0E. |
| Classe 3 | Cyclique/contrôle/Information | Adresse MAC de destination de la DLSDU |

5.2.2.4 Adresse de la source

L'adresse source doit contenir l'adresse de nœud de la DLE source.

5.2.2.5 Étiquette VLAN

L'étiquette VLAN utilise l'étiquette VLAN IEEE 802.1Q et indique le VID désigné par le type de trame (RCL, cyclique, commande et information) et le PCP désigné par la priorité de trame. Le Tableau 9 montre le format d'étiquette VLAN.

Tableau 9 – Format de l'étiquette VLAN

| Type de trame | TPID 2 octets | TCI 2 octets | | |
|---------------|------------------|-----------------|--------------|----------------|
| | | PCP 3 bit. | CFI 1 bit | VID 12 bit. |
| RCL | 0x8100. | 111 | 0 | 0xFFB. |
| Cyclique | 0x8100. | 101 | 0 | 0xFFC. |
| Commande | 0x8100. | 011 | 0 | 0xFFD. |
| Informations | 0x8100. | 001 | 0 | 0xFFE(*) |

(*): Les trames d'information peuvent utiliser d'autres VID que 0xFFE(4094) de 1 à 50 et désignent plusieurs VID.

5.2.2.6 Longueur/Type

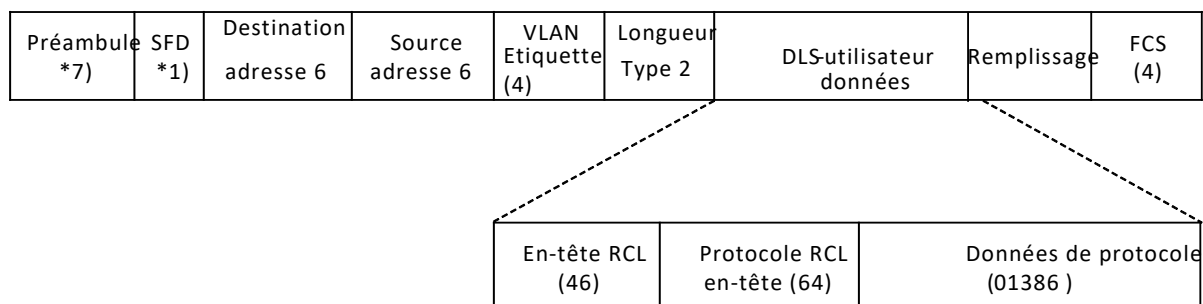
La longueur/le type est différent suivant les types de trame.

- La trame RCL est définie sur la longueur de trame. Elle est différente pour le RCL_type.
- La trame cyclique est définie sur 0x0800 (trame IP désignée).
- Les trames de commande et d'information sont mises à l'ID de protocole qui correspond aux données de PDU.

5.2.2.7 Champ Données d'utilisateur DLS

5.2.2.7.1 Trame RCL

La Figure 9 montre le format de la trame RCL. La trame RCL comporte trois segments: l'en-tête RCL, l'en-tête de protocole RCL et les données de protocole dans les données d'utilisateur DLS. Chaque segment contient des informations qui sont utilisées pour commander le réseau en anneau. Les informations diffèrent en fonction des types de trame RCL.



Légende: (#) le nombre entre parenthèses indique la longueur du champ en octets

| Anglais | Français |
|---------------------------|----------------------------------|
| Preamble (7) | Préambule (7) |
| SFD (1) | SFD (1) |
| Destination address (6) | Adresse de destination (6) |
| Source address (6) | Adresse source (6) |
| VLAN tag (4) | Étiquette VLAN (4) |
| Length/Type (2) | Longueur/Type (2) |
| DLS-user data | Données d'utilisateur DLS |
| Padding | Remplissage |
| FCS (4)' | FCS (4)' |
| RCL header (46) | En-tête RCL (46) |
| RCL protocol header (64)' | En-tête de protocole RCL (64)' |
| Protocol data' (0..1°385) | Données de protocole' (0..1°385) |

Figure 9 – Format de trame RCL

5.2.2.7.2 Trame RT

Les trames RT sont identiques à celles de l'ISO/IEC/IEEE 8802-3.

5.2.2.8 Remplissage

Un remplissage est inséré dans la trame afin que la longueur des données soit toujours un multiple de 32 bits.

5.2.2.9 Séquence de vérification de champ (FCS)

La construction, le polynôme et les résidus prévus de la séquence de contrôle de trame (FCS) sont identiques à ceux de l'ISO/IEC/IEEE 8802-3:2017, Article 3.

6 Structure spécifique à la DLPDU, codage et éléments de procédure

6.1 Généralités

L'Article 6 définit la structure, le contenu et le codage de chaque type et format de DLPDU de communication RCL et spécifie les éléments de procédure pour la DLPDU. La DLPDU de communication RT n'est pas décrite à l'Article 6, car sa trame est identique à celle de l'ISO/IEC/IEEE 8802-3.

6.2 Structure de la DLPDU de RCL

La trame de classe 1 est utilisée pour la communication avec le nœud voisin. La trame de classe 2 est utilisée pour la communication entre les nœuds autour du réseau. Pour la trame RCL, il existe une sorte de trame de classe 1 et cinq sortes de trame de classe 2. Le Tableau 10 présente les types et les classes des trames RCL.

Tableau 10 – Types et classes des trames RCL

| Classe de trame | Nom de la trame | Description |
|--|---|--|
| Classe 1 | RHE (Hello rapide) | La trame RHE confirme le nœud voisin et le chemin de transmission. |
| Classe 2 | LCC (*) (vérification de l'état de boucle) | La trame LCC est utilisée pour déterminer de manière autonome le nœud Edge-A unique, qui existe dans le réseau en anneau normal. |
| Classe 2 | LCA (*) (alerte d'état de boucle) | Si le port B d'un nœud est une liaison physique ou logique en déconnexion, la trame LCA du nœud déclare qu'il a la priorité la plus élevée pour être un nœud Edge-A. |
| Classe 2 | LCN (*) (Notification de l'état de boucle) | La trame LCN d'un nœud déclare que son propre port B passe à l'état de déconnexion de liaison physique. |
| Classe 2 | LNA (*) (Réponse de notification de la boucle) | La trame LNA est envoyée au nœud qui a envoyé la trame LNA. Seul le nœud Edge-B envoie cette trame. |
| Classe 2 | SCR (Rapport de l'état de poste) | La trame SCR notifie l'événement de changement de topologie du réseau à tous les nœuds du réseau en anneau. |
| (*)LCC, LCA, LCN et LNA ont une priorité différente pour sélectionner un nœud Edge-A: LCN > LCA > LCC (haute priorité) > LCC (basse priorité) | | |

6.2.1 En-tête RCL

6.2.1.1 Généralités

La structure de la RCL DLPDU est indiquée dans le Tableau 11.

Tableau 11 – Structure de l'en-tête RCL

| Champ d'en-tête RCL | | AWG (Octet) |
|------------------------|---------------------|-------------|
| Classe | | 2 |
| Adresse de destination | Priority (priorité) | 1 |
| | Adresse station | 1 |
| | Adresse MAC | 6 |
| Adresse de la source | Priority (priorité) | 1 |
| | Adresse station | 1 |
| | Adresse MAC | 6 |
| CMD | | 4 |
| Numéro d'ordre | | 4 |
| Réservé | | 20 |

6.2.1.2 Classe

Il existe deux types de classe dans la trame RCL (voir Tableau 12).

- Classe 1: utilisée pour la communication avec le nœud voisin;
- Classe 2: utilisée pour la communication entre les nœuds autour du réseau.

Tableau 12 – Format du champ

| Classe de trame | Valeur (2 octets) |
|-----------------|-------------------|
| Classe 1 | 0x0001. |
| Classe 2 | 0x0002. |

6.2.1.3 Adresse de destination

Le Tableau 13 montre le format du champ d'adresse de destination. Définir la valeur de priorité, l'adresse de poste, l'adresse MAC, en fonction de RCL_type.

Tableau 13 – Format du champ adresse de destination

| Point | RCL_type | |
|---------------------|-------------------|--|
| | RHE/LCC/LCN/SCR | LCA/LNA |
| Priority (priorité) | 0x00. | Défini sur l'adresse source qui reçoit la trame RCL (LCA: trame LCC reçue, LNA: trame LCN reçue) |
| Adresse station | 0xFF. | |
| Adresse MAC | FF-FF-FF-FF-FF-FF | |

6.2.1.4 Adresse de la source

Toutes les trames RCL sont positionnées sur les données indiquées dans le Tableau 14.

Tableau 14 – Format du champ Adresse de source

| Point | Valeur |
|---------------------|--------------------------|
| Priority (priorité) | 0x00. |
| Adresse station | Adresse de poste du nœud |
| Adresse MAC | Adresse MAC du nœud |

6.2.1.5 CMD

Le CMD est utilisé pour identifier la trame RCL. Cette zone diffère selon le type de trame (RCL_type) (voir Tableau 15).

Tableau 15 – Format du champ Adresse de source

| Bit | Point | Valeur | | | | | |
|----------|-----------------|--------|---|-----|-----|-----|-----|
| | | RHE | LCC (porte- puce non enfichable, Leadless Chip Carrier) | ACV | LCN | LNA | SCR |
| [3..0] | Réservé | 1 | | | | | |
| [7..4] | Numéro unique 1 | 0 | 0 | 1 | 2 | 3 | 0 |
| [11..8] | Numéro unique 2 | 0 | 1 | 1 | 1 | 1 | 3 |
| [15..12] | Réservé | 0 | | | | | |
| [23..16] | Classe de trame | 1 | 2 | 2 | 2 | 2 | 2 |
| [31..24] | Réservé | 0 | | | | | |

6.2.1.6 Numéro d'ordre

Le numéro de séquence permet de détecter la perte et la duplication des trames reçues. Sa plage de valeurs s'étend de 0x00000000 à 0xFFFFFFFF, avec un incrément de 1.

7 Eléments de procédure de DLE

7.1 Vue d'ensemble

Le paragraphe 7.2 décrit le fonctionnement de la RCLC, de la RTC, de la TRC et de la DLM.

7.2 Commande de communication RCL (RCLC)

7.2.1 Généralités

La RCLC comporte les fonctions suivantes.

- Statut de liaison de commande de la communication avec le nœud voisin à l'aide de la trame RHE;
- Statut du nœud de commande utilisant des trames RCL de classe 2;
- Envoyer et recevoir des trames RCL.

7.2.2 Définitions de primitives

7.2.2.1 Définitions de primitives entre la RCLC et l'utilisateur DLS

Le Tableau 16 donne un récapitulatif de toutes les primitives échangées entre la DLM et l'utilisateur DLS.

Tableau 16 – Primitives et paramètres de l'interface d'utilisateur DLS

| Nom de primitive | Source | Paramètres associés | Description |
|------------------|-----------------|--|--|
| DL-RCL.req | Utilisateur DLS | (in S_add, portNum Frame_pri, RCL_type, DLSDU | Transmet une trame RCL provenant de l'utilisateur DLS. La trame RCL est acheminée vers la TRC avec le paramètre de classe. |
| DL-RCL.cnf | RCLC | (out Statut | Répond à la transmission de DLSDU terminée et rend compte du statut à l'utilisateur DLS. |
| DL-RCL.ind | RCLC | (out S_add, portNum Frame_pri, RCL_type, DLSDU | Achemine une trame RCL reçue à l'utilisateur DLS. |

Les paramètres utilisés avec les primitives échangées entre la DLM et l'utilisateur DLS sont décrits dans le Tableau 17.

Tableau 17 – Paramètres utilisés avec des primitives échangées entre DLM et utilisateur DLS

| Nom de paramètre | Description |
|------------------|--|
| S_add | Le paramètre S_add spécifie l'adresse DL du fournisseur. |
| portNum | Ce paramètre spécifie le port de transmission/réception du fournisseur. |
| Frame_pri | Ce paramètre spécifie la priorité avec VLAN du réseau en anneau. |
| RCL_type | Ce paramètre spécifie le type de RCL de la trame RCL. |
| DLSDU | Ce paramètre spécifie l'information de la trame RCL. |
| Statut | Ce paramètre permet à l'utilisateur DLMS de déterminer si le DLS demandé a été remis avec succès ou s'il y a eu un échec pour une raison quelconque. La valeur de ce paramètre est la suivante: "OK - achevé avec succès" "Echec - achèvement incomplet" |

7.2.2.2 Définitions des primitives entre la RCLC et la TRC

Le Tableau 18 donne un récapitulatif de toutes les primitives échangées entre la RCLC et la TRC.

Tableau 18 – Primitives et paramètres de l'interface d'utilisateur TRC

| Nom de primitive | Source | Paramètres associés | Description |
|------------------|--------|--|--|
| RCLC-Tx.req | RCLC | D_add, S_add, Tx_port, Frame_pri, RCL_type, DLSDU | Transmet une trame RCL à la TRC avec les paramètres associés. Le RCLC attribue le paramètre Tx_port (port A, B ou les deux). |
| RCLC-Rx.ind | TRC | D_add, S_add, Rx_port, Frame_pri, RCL_type, DLSDU | Achemine une trame RCL reçue de la TRC. |
| RCLC-SetPt.req | RCLC | BLKport, Blocage | Demande de modifier le statut de blocage d'un port pour le faire passer sur la TRC. |

Les paramètres utilisés dans les primitives échangées entre l'ARPM et la DMPM sont décrits dans le Tableau 19.

Tableau 19 – Paramètres utilisés avec des primitives échangées entre la RCLC et la TRC

| Nom de paramètre | Description |
|------------------|--|
| D_add | Le paramètre D_add spécifie l'adresse DL de l'abonné. |
| S_add | Le paramètre S_add spécifie l'adresse DL du fournisseur. |
| Tx_port | Ce paramètre spécifie le port de transmission du fournisseur. |
| Rx_port | Ce paramètre spécifie le port de réception du fournisseur. |
| Frame_pri | Ce paramètre spécifie la priorité avec VLAN du réseau en anneau. |
| RCL_type | Ce paramètre spécifie le type de RCL de la trame RCL. |
| DLSDU | Ce paramètre spécifie l'information de la trame RCL. |
| BLKport | Ce paramètre spécifie le port de blocage à proprement parler. |
| Blocage | Ce paramètre spécifie le statut de blocage à proprement parler. |

7.2.2.3 Définitions des primitives entre la RCLC et la DLM

Le Tableau 20 donne un récapitulatif de toutes les primitives échangées entre la RCLC et la DLM.

Tableau 20 – Primitives et paramètres de l'interface DLM

| Nom de primitive | Source | Paramètres associés | Description |
|------------------|--------|--|--|
| RCL_STOP.ind | RCLC | RCL_Indtype | Indique un arrêt de la trame RCL à la DLM selon le diagramme d'états de commande d'anneau. |
| RCL_START.ind | RCLC | RCL_Indtype, RCL_IndDA, RCL_IndPri, RCL_IndPort | Indique un début de la trame RCL vers la DLM selon le diagramme d'états de commande d'anneau. |
| Node_ST.ind | RCLC | Node_ST, PortA_ST, PortB_ST | Indique un changement du statut du nœud ou du port, qui prend la valeur DLM d'après le diagramme d'états de commande d'anneau. |

Les paramètres utilisés dans les primitives échangées entre l'ARPM et la DMPM sont décrits dans le Tableau 21.

Tableau 21 – Paramètres utilisés avec des primitives échangées entre DLPM et DLM

| Nom de paramètre | Description |
|------------------|---|
| RCL_Indtype | Ce paramètre spécifie le type RCL des trames RCL demandées pour la DLM en raison du changement du statut de nœud. |
| RCL_IndDA | Ce paramètre spécifie l'adresse DL des trames RCL demandées pour la DLM en raison du changement du statut de nœud. |
| RCL_IndPri | Ce paramètre spécifie la priorité avec VLAN des trames RCL demandées pour la DLM en raison du changement du statut de nœud. |
| RCL_IndPort | Ce paramètre spécifie le port de transmission des trames RCL demandées pour la DLM en raison du changement du statut de nœud. |
| Node_ST | Ce paramètre contient le statut de nœud (ISL/EGA/EGB/ITM) pour la DLM en raison du changement du statut de nœud. |
| PortA_ST | Ce paramètre contient le statut du port A sur son propre nœud pour la DLM en raison du changement du statut de nœud. |
| PortB_ST | Ce paramètre contient le statut du port B sur son propre nœud pour la DLM en raison du changement du statut de nœud. |

7.2.3 Diagramme d'états RCLC

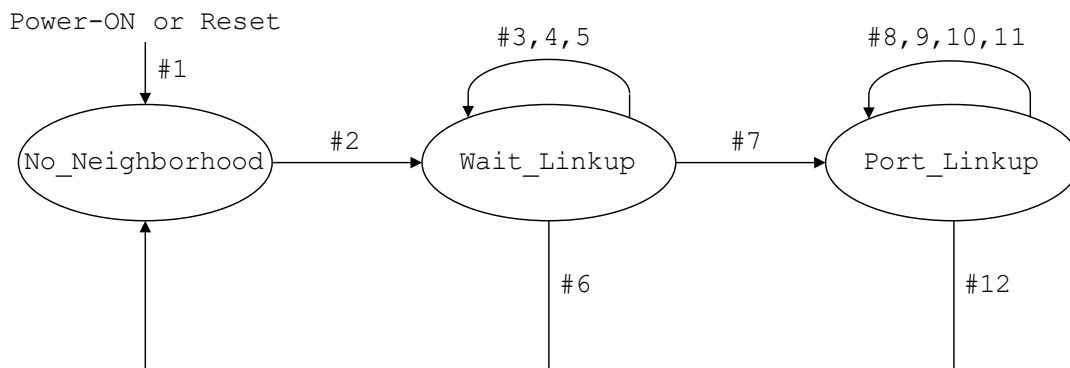
7.2.3.1 Diagramme d'états RHE du port A et B (RHE_SM-A/RHE_SM-B)

Le diagramme d'états RHE est utilisé pour confirmer le statut de liaison de la communication avec le nœud voisin. Il existe deux diagrammes d'états sur un nœud, l'un pour le port A (RHE_SM-A) et l'autre pour le port B (RHE_SM-B). L'état est mis à jour lorsque le RHE est reçu et que l'état de liaison entre le nœud et le nœud voisin est confirmé. Ces diagrammes d'états ont les états suivants.

- No_Neighborhood: Le nœud ne se connecte pas au nœud voisin ni ne reçoit la trame RHE du nœud voisin.
- Wait_Linkup: Le nœud attend l'établissement de liaisons avec le nœud voisin. Bien que cet état soit le statut d'établissement des liaisons de l'ISO/IEC/IEEE 8802-3, il convient que le nœud ne fournisse aucune trame, à l'exception de la trame RHE au nœud voisin.
- Port_Linkup: Le nœud peut communiquer avec le nœud voisin à l'aide d'une trame ISO/IEC/IEEE 8802-3 quelconque.

7.2.3.1.1 Diagramme d'états RHE_SM-A

La Figure 10 montre le diagramme d'états de RHE_SM-A et le Tableau 22 est la table d'états du port A.



| Anglais | Français |
|-------------------|-------------------|
| Power-ON or Reset | Power-ON ou Reset |
| No_Neighborhood | No_Neighborhood |
| Wait_Linkup | Wait_Linkup |
| Port_Linkup | Port_Linkup |

Figure 10 – Schéma de transition d'états de RHE_SM-A

Tableau 22 – Transitions de RHE_SM-A au niveau de la communication RCL

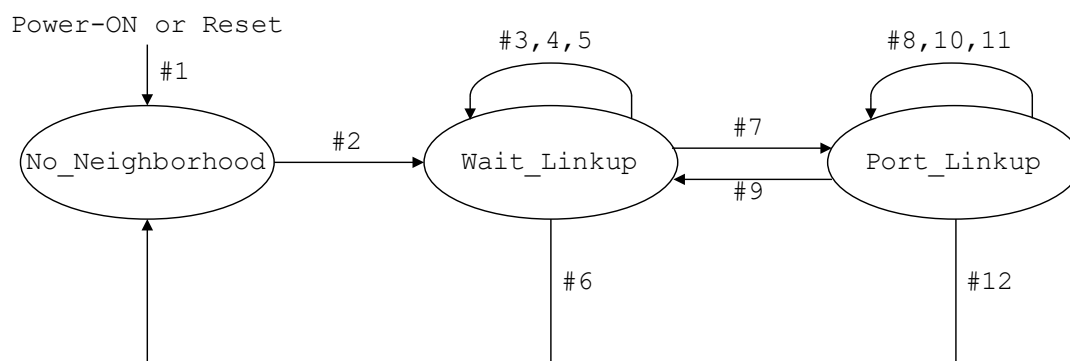
| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|---------------------|--|-----------------|
| 1 | N'importe quel état | Power-ON ou Reset => V(PA_STATUS) = NNB V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL V(PA_NBST_RCLADD) = 0 C(PA_RHE_LKUP) = 0 V(PA_RHE_RxSeq) = 0 C(PA_RHE_RxCK) = 0 START_TIMER(T(PA_RHE_TIME), V(PA_RHE_CYCLE)) | No_Neighborhood |
| 2 | No_Neighborhood | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A => V(PA_STATUS) = WLU C(PA_RHE_LKUP) = 1 C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_NBST_RCLADD) = S_add V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 3 | Wait_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && (V(PA_NBST_RCLADD) != S_add (V(PA_RHE_RxSeq)+1) != RHE_Seq) => C(PA_RHE_LKUP) = 1 C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_NBST_RCLADD) = S_add V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 4 | Wait_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && V(PA_NBST_RCLADD) == S_add && V(PA_RHE_RxSeq)+1 == RHE_Seq => C(PA_RHE_LKUP)++ C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 5 | Wait_Linkup | EXPIRED_TIMER(T(PA_RHE_TIME)) == "True" => C(PA_RHE_RxCK)-- | Wait_Linkup |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|--|-----------------|
| 6 | Wait_Linkup | /C(PA_RHE_RxCK) == 0 => V(PA_STATUS) = NNB C(PA_RHE_LKUP) = 0 V(PA_NBST_RCLADD) = 0 V(PA_RHE_RxSeq) = 0 V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL | No_Neighborhood |
| 7 | Wait_Linkup | /C(PA_RHE_LKUP) == P(RHE_LKUP_NUM) => V(PA_STATUS) = PLU C(PA_Rx-RHE_NotMatch) = 0 V(PA_NBST_LKST) = UPDATE_LKST(DLSDU) V(PA_NBST_NDST) = UPDATE_NDST(DLSDU) | Port_Linkup |
| 8 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && V(PA_NBST_RCLADD) != S_add => C(PA_Rx-RHE_NotMatch)++ | Port_Linkup |
| 9 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && V(PA_NBST_RCLADD) != S_add && C(PA_Rx-RHE_NotMatch) >= 1 => C(PA_Rx-RHE_NotMatch)++ C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_NBST_RCLADD) = S_add V(PA_NBST_LKST) = UPDATE_LKST(DLSDU) V(PA_NBST_NDST) = UPDATE_NDST(DLSDU) V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Port_Linkup |
| 10 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == A && V(PA_NBST_RCLADD) == S_add => C(PA_Rx-RHE_NotMatch) = 0 C(PA_RHE_RxCK) = P(RHE_RxCK_NUM) V(PA_NBST_LKST) = UPDATE_LKST(DLSDU) V(PA_NBST_NDST) = UPDATE_NDST(DLSDU) V(PA_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Port_Linkup |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|--|-----------------|
| 11 | Port_Linkup | EXPIRED_TIMER(T(PA_RHEcycle)) == "True" => C(PA_RHE_RxCK)-- | Port_Linkup |
| 12 | Port_Linkup | /C(PA_RHE_RxCK) == 0 => V(PA_STATUS) = NNB C(PA_RHE_LKUP) = 0 V(PA_NBST_RCLADD) = 0 V(PA_RHE_RxSeq) = 0 V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL | No_Neighborhood |

7.2.3.1.2 Diagramme d'états RHE_SM-B

La Figure 11 montre le diagramme d'états de RHE_SM-B et le Tableau 23 est la table d'états du port B.



| Anglais | Français |
|-----------------|-----------------|
| No_Neighborhood | No_Neighborhood |
| Wait_Linkup | Wait_Linkup |
| Port_Linkup | Port_Linkup |

Figure 11 – Schéma de transition d'états de RHE_SM- B

Tableau 23 – Transitions de RHE_SM-B au niveau de la communication RCL

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|---------------------|--|-----------------|
| 1 | N'importe quel état | Power-ON ou Reset => V(PB_STATUS) = NNB V(PB_NBST_LKST) = NNB V(PB_NBST_NDST) = ISL V(PB_NBST_RCLADD) = 0 C(PB_RHE_LKUP) = 0 V(PB_RHE_RxSeq) = 0 C(PB_RHE_RxCK) = 0 START_TIMER(T(PB_RHE_TIME), V(PB_RHE_CYCLE)) | No_Neighborhood |
| 2 | No_Neighborhood | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B => V(PB_STATUS) = WLU C(PB_RHE_LKUP) = 1 C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_NBST_RCLADD) = S_add V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 3 | Wait_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && (V(PB_NBST_RCLADD) != S_add V(PB_RHE_RxSeq)+1 != RHE_Seq) => C(PB_RHE_LKUP) = 1 C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_NBST_RCLADD) = S_add V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 4 | Wait_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && V(PB_NBST_RCLADD) == S_add && V(PB_RHE_RxSeq)+1 == RHE_Seq => C(PB_RHE_LKUP)++ C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 5 | Wait_Linkup | EXPIRED_TIMER(T(PB_RHE_TIME)) == "True" => C(PB_RHE_RxCK)-- | Wait_Linkup |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|--|-----------------|
| 6 | Wait_Linkup | /C(PB_RHE_RxCK) == 0 => V(PB_STATUS) = NNB C(PB_RHE_LKUP) = 0 V(PB_NBST_RCLADD) = 0 V(PB_RHE_RxSeq) = 0 V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL | No_Neighborhood |
| 7 | Wait_Linkup | /C(PB_RHE_LKUP) == P(RHE_LKUP_NUM) => V(PB_STATUS) = PLU C(PB_Rx-RHE_NotMatch) = 0 V(PB_NBST_LKST) = UPDATE_LKST(DLSDU) V(PB_NBST_NDST) = UPDATE_NDST(DLSDU) | Port_Linkup |
| 8 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && V(PB_NBST_RCLADD) != S_add => C(PB_Rx-RHE_NotMatch)++ | Port_Linkup |
| 9 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && V(PA_NBST_RCLADD) != S_add && C(PA_Rx- RHE_NotMatch) >= 1 => C(PB_Rx-RHE_NotMatch)++ C(PB_RHE_LKUP) = 1 C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_NBST_RCLADD) = S_add V(PB_NBST_LKST) = UPDATE_LKST(DLSDU) V(PB_NBST_NDST) = UPDATE_NDST(DLSDU) V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Wait_Linkup |
| 10 | Port_Linkup | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} /RCL_type == RHE && Rx_port == B && V(NBST_RCLADD) == S_add => C(PB_Rx-RHE_NotMatch) = 0 C(PB_RHE_RxCK) = P(RHE_RxCK_NUM) V(PB_NBST_LKST) = UPDATE_LKST(DLSDU) V(PB_NBST_NDST) = UPDATE_NDST(DLSDU) V(PB_RHE_RxSeq) = UPDATE_RHESeq(DLSDU) | Port_Linkup |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|--|-----------------|
| 11 | Port_Linkup | EXPIRED_TIMER(T(PB_RHEcycle)) == "True" => C(PB_RHE_RxCK)-- | Port_Linkup |
| 12 | Port_Linkup | /C(PB_RHE_RxCK) == 0 => V(PB_STATUS) = NNB C(PB_RHE_LKUP) = 0 V(PB_NBST_RCLADD) = 0 V(PB_RHE_RxSeq) = 0 V(PA_NBST_LKST) = NNB V(PA_NBST_NDST) = ISL | No_Neighborhood |

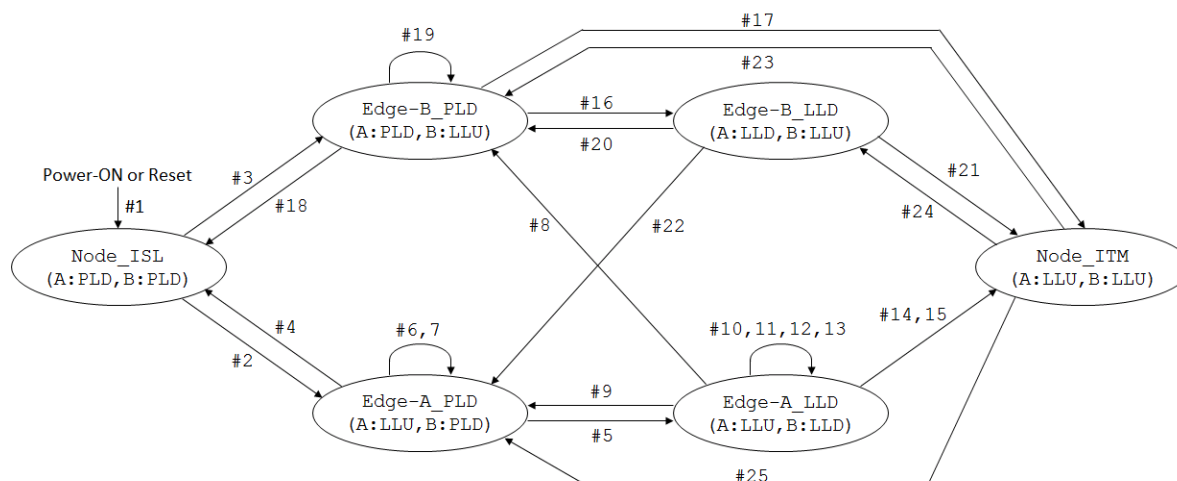
7.2.3.2 Diagramme d'états de statut de nœud RCL (RCLNode_SM)

Le RCLNode_SM sélectionne un nœud Edge-A et un nœud Edge-B dans le réseau. Tous les nœuds du réseau de Type 25 utilisent ce diagramme d'états pour décider du statut de leur propre nœud.

Le nœud contrôle le statut de liaison des ports par RCL_SM-A et RCL_SM-B. Le RCLNode_SM utilise ce statut de liaison et informe les autres nœuds du réseau par les trames RCL. Tous les nœuds du réseau sélectionnent un nœud Edge-A et un nœud Edge-B à l'aide de ce diagramme d'états. Ce diagramme d'états a les états suivants.

- Node_ISL: État de nœud isolé, les ports A et B sont bloqués.
- Edge-A_PLD État de déconnexion de liaison physique Edge-A. Dans cet état, le nœud est le nœud Edge-A, le port A est la liaison établie et le port B est bloqué en raison du facteur physique, par exemple non connecté au câble.
- Edge-A_LLD: Etat de déconnexion de liaison logique Edge-B. Dans cet état, le nœud est le nœud Edge-A, le port A est la liaison établie, le port B est bloqué et le réseau en anneau est normal.
- Edge-B_PLD: Etat de déconnexion de liaison physique Edge-B.
- Edge-B_LLD: Etat de déconnexion de liaison logique Edge-B.
- Node_ITM: Etat de nœud intermédiaire. Les ports A et B sont en liaison établie.

La Figure 12 et le Tableau 24 montrent le diagramme d'états et la table d'états de RCLNode_SM.



| Anglais | Français |
|---------------------------|---------------------------|
| Power-ON or Reset | Power-ON ou Reset |
| Node_ISL (A:PLD, B:PLD) | Node_ISL (A:PLD, B:PLD) |
| Edge-B_PLD (A:PLD, B:LLU) | Edge-B_PLD (A:PLD, B:LLU) |
| Edge-A_PLD (A:LLU, B:PLD) | Edge-A_PLD (A:LLU, B:PLD) |
| Edge-B_LLD (A:LLD, B:LLU) | Edge-B_LLD (A:LLD, B:LLU) |
| Edge-A_LLD (A:LLU, B:LLD) | Edge-A_LLD (A:LLU, B:LLD) |
| Node_ITM (A:LLU, B:LLU) | Node_ITM (A:LLU, B:LLU) |

Figure 12 – Diagramme d'états de RCLNode_SM

Tableau 24 – Transitions de RCLNode_SM au niveau de la communication RCL

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|---------------------|---|--------------|
| 1 | N'importe quel état | Power-ON ou Reset => Node_ST = ISL PortA_ST = PLD PortB_ST = PLD RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} BLKport = B RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ISL |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|--------------|---|--------------|
| 2 | Node_ISL | /V(PA_STATUS) == PLU => Node_ST = EGA PortA_ST = LLU BLKport = A Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_PLD |
| 3 | Node_ISL | /V(PB_STATUS) == PLU V(PB_NBST_NDST) != ISL => Node_ST = EGB PortB_ST = LLU BLKport = B Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_PLD |
| 4 | Edge-A_PLD | /V(PA_STATUS) == (NNB WLU) => RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} Node_ST = ISL PortA_ST = PLD BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ISL |
| 5 | Edge-A_PLD | /V(PB_STATUS) == PLU => RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} C(Rx_Counter) = 0 RCL_Indtype = LCC RCL_IndDA = Brdcast RCL_IndPri = Normal RCL_IndPort = Both RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} PortB_ST = LLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_LLD |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|--------------|--|--------------|
| 6 | Edge-A_PLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCC => RCL_Indtype = LCA RCL_IndDA = S_add RCL_IndPri = RCL_pri RCL_IndPort = Rx_port RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Edge-A_PLD |
| 7 | Edge-A_PLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LNA && D_add == MyS_add => RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} | Edge-A_PLD |
| 8 | Edge-A_LLD | /V(PA_STATUS) == (NNB WLU) => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} Node_ST = EGB PortA_ST = PLD BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} BLKport = B Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_PLD |
| 9 | Edge-A_LLD | /V(PB_STATUS) == (NNB WLU) => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} RCL_Indtype = LCN RCL_IndDA = Brdcast RCL_IndPri = V(LCN_Pri) RCL_IndPort = PortA RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} PortB_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_PLD |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|---|--------------|
| 10 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCC && RCL_IndPri < RCL_Pri => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} C(Rx_Counter) = 0 RCL_IndDA = Brdcast RCL_IndPri = Normal RCL_IndPort = Both RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Edge-A_LLD |
| 11 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCC && RCL_IndPri > RCL_Pri => RCL_Indtype = LCA RCL_IndDA = S_add RCL_IndPri = RCL_pri RCL_IndPort = Rx_port RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} C(Rx_Counter) = 0 RCL_Indtype = LCC RCL_IndDA = Brdcast RCL_IndPri = Normal RCL_IndPort = Both RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Edge-A_LLD |
| 12 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCC && RCL_IndPri == RCL_Pri => C(Rx_Counter)++ | Edge-A_LLD |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|--|--------------|
| 13 | Edge-A_LLD | /C(Rx_Counter) >= P(LCC_STOP_NUM) => C(Rx_Counter) = 0 RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_LLD |
| 14 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCA && D_add == MyS_add => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} BLKport = B Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST = ITM PortB_ST = LLU Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ITM |
| 15 | Edge-A_LLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCN => RCL_Indtype = LCC RCL_STOP.ind{RCL_Indtype} BLKport = B Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST = ITM PortB_ST = LLU Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ITM |
| 16 | Edge-B_PLD | /V(PA_STATUS) == PLU && V(PA_NBST_NDST) == EGA => PortA_ST = LLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_LLD |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|--|--------------|
| 17 | Edge-B_PLD | /V(PA_STATUS) == PLU && V(PA_NBST_NDST) == (EGB ITM) => BLKport = A Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST = ITM PortA_ST = LLU Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ITM |
| 18 | Edge-B_PLD | /V(PB_STATUS) == (NNB WLU) => RCL_Indtype = LCN RCL_STOP.ind{RCL_Indtype} BLKport = B Blocking = True RCLC-SetPT.req{BLKport,Blocking} Node_ST = ISL PortB_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ISL |
| 19 | Edge-B_PLD | RCLC-Rx.ind{D_add,S_add,Rx_port, RCL_type,RCL_Pri,DLSDU} /RCL_type == LCN => RCL_Indtype = LNA RCL_IndDA = S_add RCL_IndPri = V(LNA_Pri) RCL_IndPort = PortB RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Edge-B_PLD |
| 20 | Edge-B_LLD | /V(PA_STATUS) == (NNB WLU) => PortA_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_PLD |
| 21 | Edge-B_LLD | /V(PA_STATUS) == PLU && V(PA_NBST_NDST) == (EGB ITM) => BLKport = A Blocking = False RCLC-SetPT.req{BLKport,Blocking} Node_ST = ITM PortA_ST = LLU Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Node_ITM |

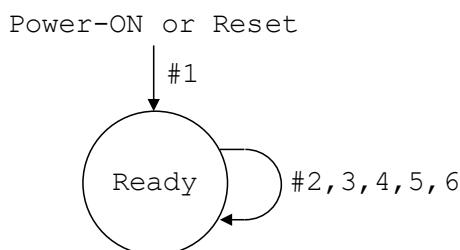
| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|--|--------------|
| 22 | Edge-B_LLD | /V(PB_STATUS) == (NNB WLU) => BLKport = B Blocking = True RCLC-SetPT.req{BLKport,Blocking} BLKport = A Blocking = False RCLC-SetPT.req{BLKport,Blocking} RCL_Indtype = LCN RCL_IndDA = Brdcast RCL_IndPri = V(LCN_pri) RCL_IndPort = PortA RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri} Node_ST = EGA PortA_ST = LLU PortB_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_PLD |
| 23 | Node_ITM | /V(PA_STATUS) == (NNB WLU) => BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} Node_ST = EGB PortA_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_PLD |
| 24 | Node_ITM | /V(PA_STATUS) == PLU && V(PA_NBST_NDST) == EGA => BLKport = A Blocking = True RCLC-SetPT.req{BLKport,Blocking} Node_ST = EGB PortB_ST = LLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-B_LLD |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|----|--------------|---|--------------|
| 25 | Node_ITM | /V(PB_STATUS) == (NNB WLU) => BLKport = B Blocking = True RCLC-SetPT.req{BLKport,Blocking} RCL_Indtype = LCN RCL_IndDA = Brdcast RCL_IndPri = V(LCN_pri) RCL_IndPort = PortA RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} Node_ST = EGA PortB_ST = PLD Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Edge-A_PLD |

7.2.3.3 Diagramme d'états de transmission/réception RCL (RCLTR_SM)

Le RCLTR_SM exécute des fonctions de transit entre l'utilisateur DLS et la TRC.

La Figure 13 montre le diagramme d'états de RCLTR_SM et le Tableau 25 est la table d'états de RCLTR_SM.



| Anglais | Français |
|-------------------|-------------------|
| Power-ON or Reset | Power-ON ou Reset |
| Ready | Prêt |

Figure 13 – Diagramme d'états de RCLTR_SM

Tableau 25 – Transitions de RCLTR_SM au niveau de la communication RCL

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|---------------------|---|--------------|
| 1 | N'importe quel état | Power-ON ou Reset => | Prêt |
| 2 | Prêt | DL-RCL.req{S_add,PortNum,Frame_pri, RCL_type,DLSDU} /RCL_type == RHE && CHK_TRCST() == "True" => Status:= "success" D_add = Class1_address Tx_port = PortNum RCLC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} DL-RCL.cnf{Status} | Prêt |
| 3 | Prêt | DL-RCL.req{S_add,PortNum,Frame_pri, RCL_type,DLSDU} /RCL_type == RHE && CHK_TRCST() != "True" => Status = "Failure" DL-RCL.cnf{Status} | Prêt |
| 4 | Prêt | DL-RCL.req{S_add,PortNum,Frame_pri, RCL_type,DLSDU} /RCL_type == (LCC LCA LCN LNA SCR) && CHK_TRCST() == "True" => Status:= "success" D_add = Class2_address Tx_port = PortNum RCLC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} DL-RCL.cnf{Status} | Prêt |
| 5 | Prêt | DL-RCL.req{S_add,PortNum,Frame_pri, RCL_type,DLSDU} /RCL_type == (LCC LCA LCN LNA SCR) && CHK_TRCST() != "True" => Status = "Failure" DL-RCL.cnf{Status} | Prêt |

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|--------------|--|--------------|
| 6 | Prêt | RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_Pri,RCL_type,DLSDU} => portNum = Rx_port DL-RCL.ind{S_add,PortNum,Frame_pri, RCL_type,DLSDU} | Prêt |

7.2.4 Fonctions de la RCLC

Toutes les fonctions utilisées par la RCLC sont résumées dans le Tableau 26.

Tableau 26 – Table des fonctions de la RCLC

| Nom de la fonction | Entrée | Sortie | Description et fonctionnement |
|--------------------|-------------|----------------------------------|---|
| UPDATE_LKST | DLSDU | Etat de la liaison (NNB/WLU/PLU) | Renvoyer le statut de liaison du nœud voisin. Le statut de liaison est inclus dans la DLSDU reçue. |
| UPDATE_NDST | DLSDU | Statut du nœud (ISL/EGA/EGB/ITM) | Renvoie le statut de nœud du nœud voisin. Le statut de nœud est inclus dans la DLSDU reçue. |
| UPDATE_RHESeq | DLSDU | 0x00000000. ~ 0xFFFFFFFF. | Renvoie 4 octets du numéro de séquence contenu dans la DLSDU. |
| START_TIMER | Tim_ID, Val | (<none>) | Le temporisateur TIM_ID est réglé par la valeur de Val et est activé. |
| EXPIRED_TIMER | TIM_ID | Vrai/Faux | Lorsque le temporisateur demandé TIM_ID a expiré, « True » est renvoyé, sinon False est renvoyé. |
| CHK_TRCST | (<none>) | Vrai/Faux | Renvoie le statut de la TRC. Si la TRC est capable d'envoyer la trame en raison de l'absence d'autres trames dans la mémoire tampon de transmission, elle renvoie « True », sinon elle renvoie « False ». |

7.3 Commande de communication en temps réel (RTC)

7.3.1 Généralités

La RTC permet d'envoyer et de recevoir les trames de communication cyclique, de commande et d'information.

7.3.2 Définitions de primitives

7.3.2.1 Définitions de primitives entre la RTC et l'utilisateur DLS

Le Tableau 27 donne un récapitulatif de toutes les primitives échangées entre la DLM et l'utilisateur DLS.

Tableau 27 – Primitives et paramètres de l'interface d'utilisateur DLS

| Nom de la primitive | Source | Paramètres associés | Description |
|---------------------|-----------------|--|---|
| DL-RTC.req | Utilisateur DLS | (in D_add, S_add, Frame_pri, DLSDU | Transmet une trame RT provenant de l'utilisateur DLS. La trame RT est acheminée vers la TRC avec les paramètres associés. |
| DL-RTC.cnf | RTC | (out Status) | Répond à la transmission de DLSDU terminée et rend compte du statut à l'utilisateur DLS. |
| DL-RTC.ind | RTC | (out D_add, S_add, Frame_pri, DLSDU | Achemine une trame RT reçue vers l'utilisateur DLS. |

Les paramètres utilisés avec les primitives échangées entre la DLM et l'utilisateur DLS sont décrits dans le Tableau 28.

Tableau 28 – Paramètres utilisés avec des primitives échangées entre DLM et utilisateur DLS

| Nom de paramètre | Description |
|------------------|--|
| D_add | Le paramètre D_add spécifie l'adresse DL de l'abonné. |
| S_add | Le paramètre S_add spécifie l'adresse DL du fournisseur. |
| Frame_pri | Ce paramètre spécifie la priorité avec VLAN du réseau en anneau. |
| DLSDU | Ce paramètre spécifie l'information de la trame RT. |
| Statut | Ce paramètre permet à l'utilisateur DLMS de déterminer si le DLS demandé a été remis avec succès ou s'il y a eu un échec pour une raison quelconque. La valeur de ce paramètre est la suivante: "OK - achevé avec succès" "Echec - achèvement incomplet" |

7.3.2.2 Définitions des primitives entre la RTC et la TRC

Le Tableau 29 donne un récapitulatif de toutes les primitives échangées entre la RTC et la TRC.

Tableau 29 – Primitives et paramètres de l'interface d'utilisateur TRC

| Nom de la primitive | Source | Paramètres associés | Description |
|---------------------|--------|---|--|
| RTC-Tx.req | RTC | D_add, S_add, Tx_port Frame_pri, DLSDU | Transmet une trame RT à la TRC avec les paramètres associés. La RTC attache le paramètre Tx_port « Both ». |
| RTC-Rx.ind | TRC | D_add, S_add, Rx_port, Frame_pri, DLSDU | Achemine une trame RT reçue de la TRC. |

Les paramètres utilisés dans les primitives échangées entre l'ARPM et la DMPM sont décrits dans le Tableau 30.

Tableau 30 – Paramètres utilisés avec des primitives échangées entre la RTC et la TRC

| Nom de paramètre | Description |
|------------------|--|
| D_add | Le paramètre D_add spécifie l'adresse DL de l'abonné. |
| S_add | Le paramètre S_add spécifie l'adresse DL du fournisseur. |
| Tx_port | Ce paramètre spécifie le port de transmission du fournisseur. |
| Rx_port | Ce paramètre spécifie le port de réception du fournisseur. |
| Frame_pri | Ce paramètre spécifie la priorité avec VLAN du réseau en anneau. |
| DLSDU | Ce paramètre spécifie l'information de la trame RT. |

7.3.2.3 Définitions des primitives entre la RTC et la DLM

Le Tableau 31 donne un récapitulatif de toutes les primitives échangées entre la RTC et la DLM.

Tableau 31 – Primitives et paramètres de l'interface DLM

| Nom de la primitive | Source | Paramètres associés | Description |
|---------------------|-------------------------------|---|---|
| RTC-Reset.ind | Gestion de liaison de données | (aucune) | Indique une réinitialisation de la RTC provenant de l'utilisateur DLS. |
| RTC-Event.ind | RTC | DLM_event identifier, Additional_information | Ce service est utilisé pour informer la DLM de certains événements ou de certaines erreurs dans la DLL. |

Les paramètres utilisés dans les primitives échangées entre l'ARPM et la DMPM sont décrits dans le Tableau 32.

Tableau 32 – Paramètres utilisés avec des primitives échangées entre DLPM et DLM

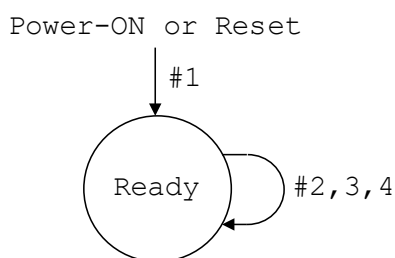
| Nom de paramètre | Description |
|------------------------|---|
| DLM_event identifier | Ce paramètre spécifie la primitive ou l'événement composite au sein de la DLE dont l'apparition est en train d'être annoncée. |
| Additional_information | Ce paramètre facultatif fournit des informations supplémentaires spécifiques à un événement. |

7.3.3 Diagramme d'états RTC

7.3.3.1 Diagramme d'états de transmission et réception RT (RTTR_SM)

Le RTTR_SM exécute des fonctions de transit entre l'utilisateur DLS et la TRC.

La Figure 14 montre le diagramme d'états de RTTR_SM et le Tableau 33 est la table d'états de RTTR_SM.



| Anglais | Français |
|-------------------|-------------------|
| Power-ON or Reset | Power-ON ou Reset |
| Ready | Prêt |

Figure 14 – Diagramme d'états de RTTR_SM

Tableau 33 – Transitions de RTTR_SM au niveau de la communication RT

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|---------------------|---|--------------|
| 1 | N'importe quel état | Power-ON ou Reset => | Prêt |
| 2 | Prêt | DL-RTC.req{D_add,S_add,Frame_pri,DLSDU} /CHK_TRCST() == "True" => Status:= "success" Tx_port = Both RTC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} DL-RTC.cnf{Status} | Prêt |
| 3 | Prêt | DL-RTC.req{D_add,S_add,Frame_pri,DLSDU} /CHK_TRCST() != "True" => Status = "Failure" DL-RTC.cnf{Status} | Prêt |
| 4 | Prêt | RTC-Rx.ind{D_add,S_add,Rx_port, Frame_Pri,DLSDU} => DL-RTC.ind{D_add,S_add,Frame_pri,DLSDU} | Prêt |

7.3.4 Fonctions de la RTC

Toutes les fonctions utilisées par la RTC sont résumées dans le Tableau 34.

Tableau 34 – Table des fonctions RTC

| Nom de la fonction | Entrée | Sortie | Description et fonctionnement |
|--------------------|----------|-----------|--|
| CHK_TRCST | (<none>) | Vrai/Faux | Renvoie le statut de la TRC. Si la TRC est capable d'envoyer la trame en raison de l'absence d'autres trames dans la mémoire tampon de transmission, elle renvoie « True », dans le cas contraire, elle renvoie « False ». |

7.4 Commande de transmission/réception (TRC)

7.4.1 Généralités

La TRC comporte les fonctions suivantes:

- Envoyer et recevoir toutes les trames du réseau de Type 25;
- Contrôler le statut du port lorsque la RCLC l'ordonne et indiquer l'état du port à la RCLC;
- Transmettre les trames reçues d'un port à un autre port.

7.4.2 Définitions de primitives

7.4.2.1 Définitions des primitives entre la TRC et la DLM

Le Tableau 35 donne un récapitulatif de toutes les primitives échangées entre la TRC et la DLM.

Tableau 35 – Primitives et paramètres de l'interface DLM

| Nom de la primitiv | Source | Paramètres associés | Description |
|--------------------|-------------------------------|---|---|
| TRC-Reset.ind | Gestion de liaison de données | (aucune) | Indique une réinitialisation de la TRC provenant de l'utilisateur DLS. |
| TRC-Event.ind | TRC | DLM_event identifier, Additional_information | Ce service est utilisé pour informer la DLM de certains événements ou de certaines erreurs dans la DLL. |

Les paramètres utilisés dans les primitives échangées entre l'ARPM et la DMPM sont décrits dans le Tableau 36.

Tableau 36 – Paramètres utilisés avec des primitives échangées entre DLPM et DLM

| Nom de paramètre | Description |
|------------------------|---|
| DLM_event identifier | Ce paramètre spécifie la primitive ou l'événement composite au sein de la DLE dont l'apparition est en train d'être annoncée. |
| Additional_information | Ce paramètre facultatif fournit des informations supplémentaires spécifiques à un événement. |

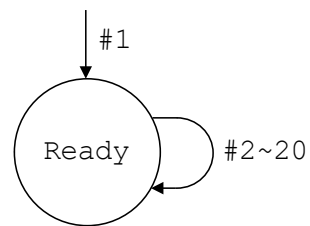
7.4.3 Diagramme d'états TRC

7.4.3.1 Diagramme d'états TRC (TRC_SM)

Le TRC_SM contrôle s'il est nécessaire d'envoyer ou de recevoir les trames RCL et RT selon le statut de blocage déterminé par la RCLC. Le TRC_SM indique le statut du port à la RCLC.

La Figure 15 montre le diagramme d'états de TRC_SM et le Tableau 37 est la table d'états de TRC_SM.

Power-ON or Reset



| Anglais | Français |
|-------------------|-------------------|
| Power-ON or Reset | Power-ON ou Reset |
| Ready | Prêt |

Figure 15 – Diagramme d'états de TRC_SM

Tableau 37 – Transitions de TRC_SM

| # | Etat actuel, | Événement /Condition => Action | état suivant |
|---|---------------------|---|--------------|
| 1 | N'importe quel état | Power-ON ou Reset => V(PA_Blocking) = True V(PB_Blocking) = True | Prêt |
| 2 | Prêt | RCLC-SetPT.req{BLKport,Blocking} /BLKport == A => V(PA_Blocking) = Blocking | Prêt |
| 3 | Prêt | RCLC-SetPT.req{BLKport,Blocking} /BLKport == B => V(PB_Blocking) = Blocking | Prêt |
| 4 | Prêt | Phs-A_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add == Class1frame => Rx_port = A RCL_type = GET_RCLTYPE(DLSDU) RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} | Prêt |
| 5 | Prêt | Phs-B_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add == Class1frame => Rx_port = B RCL_type = GET_RCLTYPE(DLSDU) RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} | Prêt |

| # | Etat actuel, | Événement /Condition => Action | état suivant |
|---|--------------|--|--------------|
| 6 | Prêt | <pre>Phs-A_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add == Class2frame && !(V(PA_Blocking) == True && V(PB_Blocking) == True) => Rx_port = A RCL_type = GET_RCLTYPE(DLSDU) RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} IF(V(PA_Blocking) == True V(PB_Blocking) == False) THEN Put_Queue-B(D_add,S_add,Frame_pri, DLSDU) ENDIF</pre> | Prêt |
| 7 | Prêt | <pre>Phs-B_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add == Class2frame && !(V(PA_Blocking) == True && V(PB_Blocking) == True) => Rx_port = B RCL_type = GET_RCLTYPE(DLSDU) RCLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,RCL_type,DLSDU} IF(V(PA_Blocking) == True V(PB_Blocking) == False) THEN Put_Queue-A(D_add,S_add,Frame_pri, DLSDU) ENDIF</pre> | Prêt |
| 8 | Prêt | <pre>Phs-A_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add != Class1frame && D_add != Class2frame && V(PA_Blocking) == False => Rx_port = A RLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,DLSDU} IF(V(PB_Blocking) == False) THEN Put_Queue-B(D_add,S_add,Frame_pri, DLSDU) ENDIF</pre> | Prêt |

| # | Etat actuel, | Événement /Condition => Action | état suivant |
|----|--------------|--|--------------|
| 9 | Prêt | <pre> Phs-B_Data.ind{D_add,S_add, Frame_pri,DLSDU} /D_add != Class1frame && D_add != Class2frame && V(PB_Blocking) == False => Rx_port = B RLC-Rx.ind{D_add,S_add,Rx_port, Frame_pri,DLSDU} IF(V(PA_Blocking) == False) THEN Put_Queue-A(D_add,S_add,Frame_pri, DLSDU ENDIF </pre> | Prêt |
| 10 | Prêt | <pre> RCLC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} /D_add == Class1frame => IF(Tx_port == A) THEN Put_Queue-A(D_add,S_add,Frame_pri, DLSDU ENDIF IF(Tx_port == B) THEN PUT_Queue-B(D_add,S_add,Frame_pri, DLSDU ENDIF </pre> | Prêt |
| 11 | Prêt | <pre> RCLC-Tx.req{D_add,S_add,Tx_port, Frame_pri,RCL_type,DLSDU} /D_add == Class2frame && !(V(PA_Blocking) == True && V(PB_Blocking) == True) => IF(Tx_port == Both Tx_port == A) THEN Put_Queue-A(D_add,S_add,Frame_pri, DLSDU IF(Tx_port == Both Tx_port == B) THEN Put_Queue-B(D_add,S_add,Frame_pri, DLSDU </pre> | Prêt |

| # | Etat actuel, | Événement /Condition => Action | état suivant |
|----|--------------|--|--------------|
| 12 | Prêt | RTC-Tx.req{D_add,S_add,Tx_port, Frame_pri,DLSDU} /D_add != Class1frame && D_add != Class2frame && Tx_port == Both => IF(V(PA_Blocking) == False) THEN Put_Queue-A(D_add,S_add, Frame_pri,DLSDU) ENDIF IF(V(PB_Blocking) == False) THEN Put_Queue-B(D_add,S_add, Frame_pri,DLSDU) ENDIF | Prêt |
| 13 | Prêt | /Queue-A_Check(VLAN_RCL) == True => Send_Quuedata-A(VLAN_RCL) | Prêt |
| 14 | Prêt | /Queue-B_Check(VLAN_RCL) == True => Send_Quuedata-B(VLAN_RCL) | Prêt |
| 15 | Prêt | /Queue-A_Check(VLAN_CYC) == True && Queue-A_Check(VLAN_RCL) == False => Send_Quuedata-A(VLAN_CYC) | Prêt |
| 16 | Prêt | /Queue-B_Check(VLAN_CYC) == True && Queue-B_Check(VLAN_RCL) == False => Send_Quuedata-B(VLAN_CYC) | Prêt |
| 17 | Prêt | /Queue-A_Check(VLAN_CTL) == True && Queue-A_Check(VLAN_RCL) == False && Queue-A_Check(VLAN_CYC) == False => Send_Quuedata-A(VLAN_CTL) | Prêt |
| 18 | Prêt | /Queue-B_Check(VLAN_CTL) == True && Queue-B_Check(VLAN_RCL) == False && Queue-B_Check(VLAN_CYC) == False => Send_Quuedata-B(VLAN_CTL) | Prêt |

| # | Etat actuel, | Événement /Condition => Action | état suivant |
|----|--------------|--|--------------|
| 19 | Prêt | /Queue-A_Check(VLAN_INFO) == True && Queue-A_Check(VLAN_RCL) == False && Queue-A_Check(VLAN_CYC) == False && Queue-A_Check(VLAN_CTL) == False => Send_Queueudata-A(VLAN_INFO) | Prêt |
| 20 | Prêt | /Queue-B_Check(VLAN_INFO) == True && Queue-B_Check(VLAN_RCL) == False && Queue-B_Check(VLAN_CYC) == False && Queue-B_Check(VLAN_CTL) == False => Send_Queueudata-B(VLAN_INFO) | Prêt |

7.4.4 Fonctions de la TRC

Toutes les fonctions utilisées par la TRC sont résumées dans le Tableau 38.

Tableau 38 – Table des fonctions de la TRC

| Nom de la fonction | Entrée | Sortie | Description et fonctionnement |
|--------------------|---|---|--|
| GET_RCLTYPE | DLSDU | RCL_type (RHE/LCC/ LCA/LCN/ LNA/SCR) | Renvoie le RCL_type inclus dans la DLSDU. |
| Put_Queue-A | D_add, S_add, Frame_pri, DLSDU | (<none>) | Met la PDU d'entrée en file d'attente de transmission dans le port A en fonction de Frame_pri selon un mode FIFO. |
| Put_Queue-B | D_add, S_add, Frame_pri, DLSDU | (<none>) | Met la PDU d'entrée en file d'attente de transmission dans le port B en fonction de Frame_pri selon un mode FIFO. |
| Queue-A_Check | Frame_pri | Vrai/Faux | Renvoie « True » si la file d'attente du port A spécifiée par Frame_pri n'est pas vide. |
| Queue-B_Check | Frame_pri | Vrai/Faux | Renvoie « True » si la file d'attente du port B spécifiée par Frame_pri n'est pas vide. |
| Send_Queueudata-A | Frame_pri | (<none>) | Se retire de la file d'attente du port A spécifiée par Frame_pri selon le mode FIFO et envoie la PDU depuis le port A. Send_Queueudata-A(Frame_pri) est assemblé comme suit: portNum = PortA D_add = dequeue_Dadd(PortNum, Frame_Pri) S_add = dequeue_Sadd(PortNum, Frame_Pri) DLSDU = dequeue_data(PortNum, Frame_Pri) Phs-A_Data.req{D_add,S_add, Frame_pri,DLSDU} |

| Nom de la fonction | Entrée | Sortie | Description et fonctionnement |
|--------------------|-----------------------|----------|---|
| Send_QueueData-B | Frame_pri | (<none>) | Se retire de la file d'attente du port A spécifiée par Frame_pri selon le mode FIFO et envoie la PDU depuis le port A. Send_QueueData-B(Frame_pri) est assemblé comme suit: PortNum = PortB D_add = queue_Dadd(PortNum, Frame_Pri) S_add = queue_Sadd(PortNum, Frame_Pri) DLSDU = queue_data(PortNum, Frame_Pri) Phs-B_Data.req{D_add,S_add, Frame_pri,DLSDU} |
| dequeue_Dadd | PortNum, Frame_Pri | D_add | Retire l'adresse de destination de la file d'attente spécifiée par PortNum et Frame_Pri selon le mode FIFO. |
| dequeue_Sadd | PortNum, Frame_Pri | S_add | Retire l'adresse source de la file d'attente spécifiée par PortNum et Frame_Pri selon le mode FIFO. |
| dequeue_data | PortNum, Frame_Pri | DLSDU | Retire la DLSDU de la file d'attente spécifiée par PortNum et Frame_Pri selon un mode FIFO. |

7.5 Protocole de gestion DLL (DLM)

7.5.1 Vue d'ensemble

Le protocole d'interface entre la DLM et l'utilisateur DLMS est décrit en 7.5.

7.5.2 Définitions de primitives

7.5.2.1 Primitives échangées entre DLM et utilisateur DLS

Le Tableau 39 donne un récapitulatif de toutes les primitives échangées entre la DLM et l'utilisateur DLS.

Tableau 39 – Primitives échangées entre DLM et utilisateur DLS

| Nom de la primitive | Source | Paramètres associés | Description |
|---------------------|-------------------------------|--|---|
| DLM_RCL_STOP.ind | Gestion de liaison de données | (out RCL_Indtype) | Indique un arrêt de la trame RCL à l'utilisateur DLS selon le diagramme d'états de commande d'anneau. |
| DLM_RCL_START.ind | Gestion de liaison de données | (out RCL_Indtype, RCL_IndDA, RCL_IndPri, RCL_IndPort) | Indique un début de la trame RCL à l'utilisateur DLS selon le diagramme d'états de commande d'anneau. |
| DLM_Node_ST.ind | Gestion de liaison de données | (out Node_ST, PortA_ST, PortB_ST) | Indique un changement de statut du nœud ou du port à l'utilisateur DLS selon le diagramme d'états de commande d'anneau. |

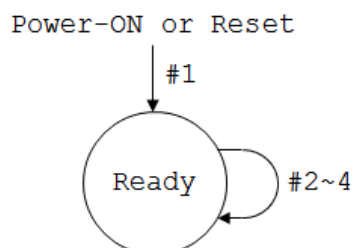
Les paramètres utilisés avec les primitives échangées entre la DLM et l'utilisateur DLS sont décrits dans le Tableau 40.

Tableau 40 – Paramètres utilisés avec des primitives échangées entre DLM et utilisateur DLS

| Nom de paramètre | Description |
|------------------|---|
| RCL_Indtype | Ce paramètre spécifie le type RCL des trames RCL demandées pour la DLM en raison du changement du statut de nœud. |
| RCL_IndDA | Ce paramètre spécifie l'adresse DL des trames RCL demandées pour la DLM en raison du changement du statut de nœud. |
| RCL_IndPri | Ce paramètre spécifie la priorité avec VLAN des trames RCL demandées pour la DLM en raison du changement du statut de nœud. |
| RCL_IndPort | Ce paramètre spécifie le port de transmission des trames RCL demandées pour la DLM en raison du changement du statut de nœud. |
| Node_ST | Ce paramètre contient le statut de nœud (ISL/EGA/EGB/ITM) pour la DLM en raison du changement du statut de nœud. |
| PortA_ST | Ce paramètre contient le statut du port A sur son propre nœud pour la DLM en raison du changement du statut de nœud. |
| PortB_ST | Ce paramètre contient le statut du port B sur son propre nœud pour la DLM en raison du changement du statut de nœud. |

7.5.3 Diagramme d'états DLM (DLM_SM)

La Figure 16 montre le diagramme d'états de DLM_SM et le Tableau 41 est la table d'états de DLM_SM.



| Anglais | Français |
|-------------------|-------------------|
| Power-ON or Reset | Power-ON ou Reset |
| Ready | Prêt |

Figure 16 – Diagramme d'états de DLM_SM

Tableau 41 – Transitions de DLM_SM

| # | Etat actuel, | Événement /Condition => Action | Etat suivant |
|---|---------------------|---|--------------|
| 1 | N'importe quel état | Power-ON ou Reset => | Prêt |
| 2 | Prêt | RCL_STOP.ind{RCL_Indtype} / => DLM_RCL_STOP.ind{RCL_Indtype} | Prêt |
| 3 | Prêt | RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} / => DLM_RCL_START.ind{RCL_Indtype,RCL_IndDA, RCL_IndPri,RCL_IndPort} | Prêt |
| 4 | Prêt | Node_ST.ind{Node_ST,PortA_ST,PortB_ST} / => DLM_Node_ST.ind{Node_ST,PortA_ST,PortB_ST} | Prêt |

Bibliographie

IEC 61158-1:2019, *Réseaux de communication industriels - Spécifications des bus de terrain - Partie 1 : présentation et lignes directrices des séries IEC 61158 et IEC 61784*

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

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