

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE



**Industrial communication networks – High availability automation networks –  
Part 7: Ring-based Redundancy Protocol (RRP)**

**Réseaux de communication industriels – Réseaux de haute disponibilité pour  
l'automation –  
Partie 7: Protocole de redondance pour réseau en anneau (RRP)**

IECNORM.COM : Click to view the full PDF of IEC 62439-7:2011



## THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2011 IEC, Geneva, Switzerland

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

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

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

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

IEC Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland  
Email: [inmail@iec.ch](mailto:inmail@iec.ch)  
Web: [www.iec.ch](http://www.iec.ch)

## About the IEC

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

## About IEC publications

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

- Catalogue of IEC publications: [www.iec.ch/searchpub](http://www.iec.ch/searchpub)

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: [www.iec.ch/online\\_news/justpub](http://www.iec.ch/online_news/justpub)

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

- Electropedia: [www.electropedia.org](http://www.electropedia.org)

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

- Customer Service Centre: [www.iec.ch/webstore/custserv](http://www.iec.ch/webstore/custserv)

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

Email: [csc@iec.ch](mailto:csc@iec.ch)

Tel.: +41 22 919 02 11

Fax: +41 22 919 03 00

## A propos de la CEI

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

## A propos des publications CEI

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

- Catalogue des publications de la CEI: [www.iec.ch/searchpub/cur\\_fut-f.htm](http://www.iec.ch/searchpub/cur_fut-f.htm)

Le Catalogue en-ligne de la CEI vous permet d'effectuer des recherches en utilisant différents critères (numéro de référence, texte, comité d'études,...). Il donne aussi des informations sur les projets et les publications retirées ou remplacées.

- Just Published CEI: [www.iec.ch/online\\_news/justpub](http://www.iec.ch/online_news/justpub)

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

- Electropedia: [www.electropedia.org](http://www.electropedia.org)

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International en ligne.

- Service Clients: [www.iec.ch/webstore/custserv/custserv\\_entry-f.htm](http://www.iec.ch/webstore/custserv/custserv_entry-f.htm)

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions, visitez le FAQ du Service clients ou contactez-nous:

Email: [csc@iec.ch](mailto:csc@iec.ch)

Tél.: +41 22 919 02 11

Fax: +41 22 919 03 00

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE



**Industrial communication networks – High availability automation networks –  
Part 7: Ring-based Redundancy Protocol (RRP)**

**Réseaux de communication industriels – Réseaux de haute disponibilité pour  
l'automatisation –  
Partie 7: Protocole de redondance pour réseau en anneau (RRP)**

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

COMMISSION  
ELECTROTECHNIQUE  
INTERNATIONALE

PRICE CODE  
CODE PRIX

**XC**

ICS 25.040; 35.040

ISBN 978-2-88912-838-9

## CONTENTS

|                                                                      |    |
|----------------------------------------------------------------------|----|
| FOREWORD.....                                                        | 6  |
| INTRODUCTION.....                                                    | 8  |
| 1 Scope.....                                                         | 9  |
| 2 Normative references .....                                         | 9  |
| 3 Terms, definitions, abbreviations, acronyms, and conventions ..... | 9  |
| 3.1 Terms and definitions .....                                      | 9  |
| 3.2 Abbreviations and acronyms.....                                  | 10 |
| 3.3 Conventions .....                                                | 10 |
| 4 RRP overview.....                                                  | 11 |
| 4.1 General.....                                                     | 11 |
| 4.2 Frame forwarding and receiving control .....                     | 11 |
| 4.2.1 General .....                                                  | 11 |
| 4.2.2 Normal Device (ND) and Gateway Device (GWD) .....              | 11 |
| 4.2.3 Behaviours of the General Device (GD) .....                    | 12 |
| 4.2.4 Behaviours of the Line Network Manager (LNM).....              | 13 |
| 4.2.5 Behaviours of the Ring Network Managers (RNMs) .....           | 13 |
| 4.3 Link status monitoring .....                                     | 14 |
| 4.4 Error detection .....                                            | 14 |
| 4.5 Plug and play .....                                              | 14 |
| 4.6 Network management information base (NMIB) management .....      | 14 |
| 4.7 Network recovery .....                                           | 15 |
| 4.8 Automatic network configuration.....                             | 15 |
| 4.9 RRP basic operating principle .....                              | 15 |
| 5 RRP redundancy behaviours .....                                    | 17 |
| 5.1 Network topology.....                                            | 17 |
| 5.2 Network recovery in ring network.....                            | 18 |
| 5.2.1 General .....                                                  | 18 |
| 5.2.2 Link fault between neighbouring devices .....                  | 20 |
| 5.2.3 Link fault of remote device.....                               | 21 |
| 5.2.4 Device fault on a RNM.....                                     | 22 |
| 5.3 Automatic Ring Network Manager (RNM) election procedure.....     | 23 |
| 5.3.1 General .....                                                  | 23 |
| 5.3.2 Primary RNM (RNMP).....                                        | 23 |
| 5.3.3 Secondary RNM (RNMS) .....                                     | 24 |
| 5.4 Path management .....                                            | 24 |
| 5.4.1 General .....                                                  | 24 |
| 5.4.2 Path in a line topology network.....                           | 24 |
| 5.4.3 Path in a ring topology network.....                           | 25 |
| 5.5 Device address collision .....                                   | 26 |
| 6 RRP class specification .....                                      | 27 |
| 6.1 General .....                                                    | 27 |
| 6.2 Template .....                                                   | 27 |
| 6.3 Attributes.....                                                  | 29 |
| 7 RRP services specification .....                                   | 34 |
| 7.1 Set device information.....                                      | 34 |
| 7.2 Get device information .....                                     | 36 |

|       |                                                               |    |
|-------|---------------------------------------------------------------|----|
| 7.3   | Get network information .....                                 | 38 |
| 7.4   | Get path table information .....                              | 40 |
| 8     | RRP protocol specification.....                               | 42 |
| 8.1   | General .....                                                 | 42 |
| 8.2   | Ethernet header .....                                         | 43 |
| 8.2.1 | Preamble.....                                                 | 43 |
| 8.2.2 | Start frame delimiter .....                                   | 43 |
| 8.2.3 | Destination MAC address .....                                 | 43 |
| 8.2.4 | Source MAC address .....                                      | 43 |
| 8.2.5 | Length/Type .....                                             | 43 |
| 8.3   | Encoding of RRP_FrameHDR.....                                 | 43 |
| 8.3.1 | Version and length .....                                      | 43 |
| 8.3.2 | DST_addr.....                                                 | 44 |
| 8.3.3 | SRC_addr.....                                                 | 45 |
| 8.3.4 | Frame Control (FC) .....                                      | 45 |
| 8.4   | Encoding of data and pad .....                                | 47 |
| 8.4.1 | General .....                                                 | 47 |
| 8.4.2 | Encoding of FamilyReq.....                                    | 47 |
| 8.4.3 | Encoding of FamilyRes .....                                   | 48 |
| 8.4.4 | Encoding of MediaLinked.....                                  | 48 |
| 8.4.5 | Encoding of AdvThis.....                                      | 49 |
| 8.4.6 | Encoding of LineStart .....                                   | 49 |
| 8.4.7 | Encoding of RingStart.....                                    | 50 |
| 8.4.8 | Encoding of AckRNMS .....                                     | 51 |
| 8.4.9 | Encoding of CheckRNMS.....                                    | 52 |
| 8.5   | Frame Check Sequence (FCS).....                               | 52 |
| 9     | RRP protocol machine .....                                    | 52 |
| 9.1   | Protocol state machine description .....                      | 52 |
| 9.2   | Local parameters and variables for protocol state.....        | 54 |
| 9.2.1 | General .....                                                 | 54 |
| 9.2.2 | Variables to support local device information management..... | 54 |
| 9.2.3 | Variables to support network information management .....     | 55 |
| 9.2.4 | Variables to support device path information management ..... | 55 |
| 9.2.5 | Variables of Received RRP Frame.....                          | 55 |
| 9.2.6 | Local variables for protocol state .....                      | 56 |
| 9.2.7 | Constants for protocol state.....                             | 56 |
| 9.3   | State transitions .....                                       | 57 |
| 9.4   | Function descriptions .....                                   | 70 |
| 10    | RRP Management Information Base (MIB) .....                   | 75 |
|       | Bibliography.....                                             | 81 |
|       | Figure 1 – Forwarding and receiving Ethernet frames .....     | 11 |
|       | Figure 2 – Structures of ND and GWD .....                     | 12 |
|       | Figure 3 – LNM forwarding control .....                       | 13 |
|       | Figure 4 – RNM forwarding control.....                        | 13 |
|       | Figure 5 – Link status information .....                      | 14 |
|       | Figure 6 – A device operation in initialization phase.....    | 15 |

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| Figure 7 – Devices operation in line network establishing phase ..... | 16 |
| Figure 8 – Extension of line network operation.....                   | 16 |
| Figure 9 – Ring network establishment operation.....                  | 17 |
| Figure 10 – Ring to line network change operation.....                | 17 |
| Figure 11 – Ring topology .....                                       | 18 |
| Figure 12 – Link fault between neighbouring devices .....             | 21 |
| Figure 13 – Link fault of remote device .....                         | 22 |
| Figure 14 – Device fault on a RNM .....                               | 22 |
| Figure 15 – Path management in a line topology network.....           | 24 |
| Figure 16 – Path management in a ring topology network .....          | 25 |
| Figure 17 – RRP device address collision in a ring network .....      | 26 |
| Figure 18 – Common MAC frame format for RRP DLPDU .....               | 43 |
| Figure 19 – RRP protocol state machine .....                          | 53 |
| <br>                                                                  |    |
| Table 1 – RRP network recovery parameter .....                        | 19 |
| Table 2 – Parameters for calculation.....                             | 19 |
| Table 3 – Path table of Device1 in a line topology network.....       | 24 |
| Table 4 – Path table of Device4 in a line topology network.....       | 25 |
| Table 5 – Path table of Device1 in a ring topology network .....      | 25 |
| Table 6 – Path table of Device3 in a ring topology network .....      | 26 |
| Table 7 – Device address collision information.....                   | 27 |
| Table 8 – Parameters of set device information service .....          | 34 |
| Table 9 – Parameters of get device information service .....          | 36 |
| Table 10 – Parameters of get network information service .....        | 38 |
| Table 11 – Parameters of get path table information service .....     | 40 |
| Table 12 – RRP Length/Type field .....                                | 43 |
| Table 13 – Version.....                                               | 44 |
| Table 14 – DST_addr.....                                              | 44 |
| Table 15 – SRC_addr.....                                              | 45 |
| Table 16 – Network control message type.....                          | 45 |
| Table 17 – Type of service.....                                       | 45 |
| Table 18 – Priority .....                                             | 46 |
| Table 19 – Validation of extension code.....                          | 46 |
| Table 20 – Encoding of FamilyReq frame.....                           | 47 |
| Table 21 – Encoding of FamilyRes frame.....                           | 48 |
| Table 22 – Encoding of MediaLinked frame.....                         | 48 |
| Table 23 – Encoding of AdvThis frame.....                             | 49 |
| Table 24 – Encoding of LineStart frame .....                          | 49 |
| Table 25 – Encoding of RingStart frame.....                           | 50 |
| Table 26 – Encoding of AckRNMS frame .....                            | 51 |
| Table 27 – Encoding of CheckRNMS frame .....                          | 52 |
| Table 28 – Variables to support device information management .....   | 54 |
| Table 29 – Variables to support managing network information .....    | 55 |

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| Table 30 – Variables to support device path information management ..... | 55 |
| Table 31 – Variables of Received RRP Frame .....                         | 55 |
| Table 32 – Local variables for protocol state .....                      | 56 |
| Table 33 – Constants for protocol state .....                            | 56 |
| Table 34 – RRP State transitions .....                                   | 57 |
| Table 35 – RRP Function descriptions .....                               | 70 |

IECNORM.COM : Click to view the full PDF of IEC 62439-7:2011

# INTERNATIONAL ELECTROTECHNICAL COMMISSION

## INDUSTRIAL COMMUNICATION NETWORKS – HIGH AVAILABILITY AUTOMATION NETWORKS –

### Part 7: Ring-based Redundancy Protocol (RRP)

#### 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 62439-7 has been prepared by subcommittee 65C: Industrial Networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

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

|              |                  |
|--------------|------------------|
| FDIS         | Report on voting |
| 65C/668/FDIS | 65C/673/RVD      |

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

This International Standard is to be read in conjunction with IEC 62439-1:2010, *Industrial communication networks – High availability automation networks – Part 1: General concepts and calculation methods*.

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

A list of all the parts of the IEC 62439 series, under the general title *Industrial communication networks – High availability automation networks*, can be found on the IEC web site.

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

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

The contents of the corrigendum of May 2015 have been included in this copy.

**IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.**

IECNORM.COM : Click to view the full PDF of IEC 62439-7:2011

## INTRODUCTION

The IEC 62439 series specifies relevant principles for high availability networks that meet the requirements for industrial automation networks.

In the fault-free state of the network, the protocols of the IEC 62439 series provide ISO/IEC 8802-3:2000 (IEEE 802.3) with compatible, reliable data communications, and preserve determinism in real-time data communications. In cases of fault, removal, and insertion of a component, they provide deterministic recovery times.

These protocols retain fully the Ethernet communication capabilities typically used in the office world, to ensure that software that relies on these protocols will remain applicable.

The market is in need of several network solutions, each with different performance characteristics and functional capabilities, meeting diverse application requirements. These solutions support different redundancy topologies and mechanisms, which are introduced in IEC 62439-1 and specified in the companion International Standards. IEC 62439-1 also distinguishes between these different solutions, providing guidance for the user.

The IEC 62439 series follows the general structure and terms of IEC 61158 series.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning IEC 61158-4-21 given in Clause 4 and Clause 5.

Patent Number KR 0789444 "COMMUNICATION PACKET PROCESSING APPARATUS AND METHOD FOR RING TOPOLOGY ETHERNET NETWORK CAPABLE OF PREVENTING PERMANENT PACKET LOOPING," owned by LS INDUSTRIAL SYSTEMS CO., LTD., Anyang, Korea

Patent Number KR 0732510 "NETWORK SYSTEM" owned by LS INDUSTRIAL SYSTEMS CO., LTD., Anyang, Korea

Patent Number KR 0870670 "Method For Determining a Ring Manager Node", owned by LS INDUSTRIAL SYSTEMS CO., LTD., Anyang, Korea

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

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

LSIS Co Ltd  
LS Tower  
1026-6, Hogye-Dong  
Dongan-Gu  
Anyang, Gyeonggi-Do, 431-848  
South Korea  
Phone +82 2 2034 4917  
Fax +82 2 2034 4648

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](http://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 – HIGH AVAILABILITY AUTOMATION NETWORKS –

## Part 7: Ring-based Redundancy Protocol (RRP)

### 1 Scope

The IEC 62439 series of standards is applicable to high-availability automation networks based on the ISO/IEC 8802-3:2000 (Ethernet) technology.

This part of the IEC 62439 series specifies a redundancy protocol that is based on a ring topology, in which the redundancy protocol is executed at the end nodes, as opposed to being built into the switches. Each node detects link failure and link establishment using media-sensing technologies, and shares the link information with the other nodes, to guarantee fast connectivity recovery times. The nodes have equal RRP network management functions.

### 2 Normative references

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

IEC 60050-191, *International Electrotechnical Vocabulary – Chapter 191 : Dependability and quality of service*

IEC 62439-1:2010, *Industrial communication networks – High availability automation networks – Part 1: General concepts and calculation methods*

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

### 3 Terms, definitions, abbreviations, acronyms, and conventions

#### 3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 60050-191 as well as in IEC 62439-1, and the following apply.

##### 3.1.1

##### **R-port**

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

##### 3.1.2

##### **device address**

2 octet address that designates the device associated with a single device on a specific local link

### 3.1.3

#### **Gateway Device**

##### **GWD**

RRP device that has more than 3 Ethernet ports on it. At least 2 ports have to support RRP protocol

### 3.1.4

#### **Normal Device**

##### **ND**

normal RRP device which has two RRP ports on it

### 3.1.5

#### **Unique Identification**

##### **UID**

Unique 8 octet identification used to identify a RRP device within a network segment. UID is combines a 2 octet device address and a 6 octet MAC address, so that it has a unique value in a network

## **3.2 Abbreviations and acronyms**

For the purposes of this document, the abbreviations and acronyms given in IEC 62439-1 as well as the following, apply.

|      |                                     |
|------|-------------------------------------|
| ASE  | Application Service Element         |
| DLE  | Data Link layer Entity              |
| FC   | Frame Control                       |
| FCS  | Frame Check Sequence                |
| GD   | General Device                      |
| GWD  | Gateway Device                      |
| LNМ  | Line Network Manager                |
| MAC  | Media Access Control                |
| MIB  | Management Information Base         |
| NCM  | Network Control Message             |
| NCMT | Network Control Message Type        |
| ND   | Normal Device                       |
| NMIB | Network Management Information Base |
| PHY  | Physical Interface Transceiver      |
| PO   | Power On                            |
| PRI  | Priority                            |
| RES  | Reserved                            |
| RNM  | Ring Network Manager                |
| RRP  | Ring based Redundancy Protocol      |
| SA   | Stand Alone                         |
| ToS  | Type of Service                     |
| VoE  | Validation of Extension code        |

## **3.3 Conventions**

This part of the IEC 62439 series follows the conventions defined in IEC 62439-1.

## 4 RRP overview

### 4.1 General

The RRP specifies a recovery protocol, based on a ring topology. All links in an RRP network shall be full duplex through the use of an internal hardware Ethernet switch. Thus, RRP provides a collision-free transmission mechanism between two nodes. Every RRP device detects link failure and link establishment using the rules specified in ISO/IEC 8802-3:2000 and shares this information with other RRP devices so that fast connectivity recovery time is also guaranteed in the ring network.

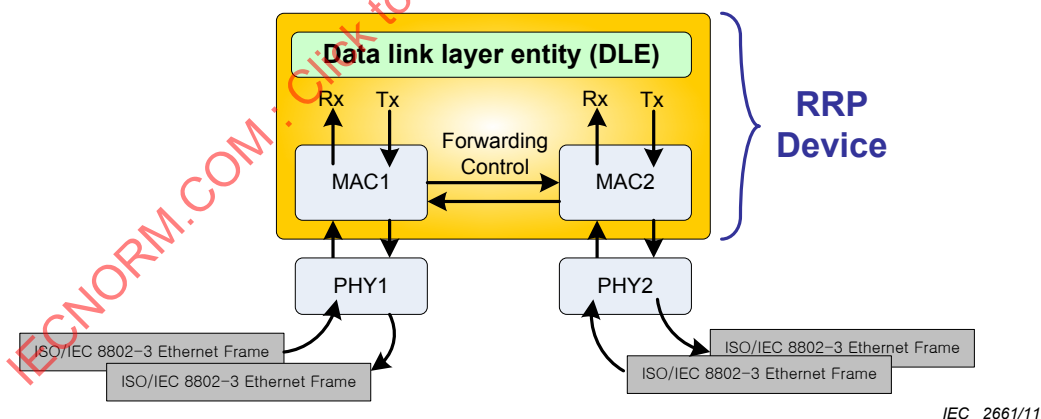
A RRP device is a dual-port switching device that receives and transmits standard ISO/IEC 8802-3:2000 Ethernet frames. It is intelligent and can control directional frame forwarding between its dual ports according to the network status and device status. RRP uses a special network management scheme specified in this standard. RRP also uses a network control based on device address and MAC address, and thus general bridge hub or switch might not be suitable for RRP network. However, when connecting a general Ethernet device to RRP network, Gateway Device (GWD) should be used.

### 4.2 Frame forwarding and receiving control

#### 4.2.1 General

RRP provides a collision-free transmission mechanism with an internal full-duplex hardware switch with switching queue and dual MACs in a device. The switching priority method between Tx and Forwarding can be Round-Robin, Tx-First or Forwarding-First scheme. However RRP does not specify the switching method.

Thus, a RRP device transmits frames without the restriction of medium access, as soon as they appear in the transmit queue for each MAC. Figure 1 shows the forwarding and receiving control of the RRP device.



IEC 2661/11

Figure 1 – Forwarding and receiving Ethernet frames

#### 4.2.2 Normal Device (ND) and Gateway Device (GWD)

RRP is operated in a dual-port ring topology. A general Ethernet device can send standard Ethernet frames through RRP ring network with GWD. Multi-ring network can also be established using GWD.

GWD is responsible for switching Ethernet frames between RRP network and external Ethernet networks through application layer using a dynamic table. The dynamic table maps addresses to external Ethernet ports automatically. The dynamic table is automatically made by learning frame movements in the network. The GWD inspects both the destination and the source addresses. The destination address is used for the forwarding decision; the source address is used for adding entries to the table and for updating purposes. When an Ethernet

frame is received at the media access control (MAC) layer through the physical interface transceiver (PHY), a GWD handles the received frame by taking one of the following actions, depending on the destination MAC address and the source MAC addresses in the received frame:

- for a broadcast or multicast frame, accept and deliver the frame to the data link layer entity (DLE), and forward the frame to the other RRP port and external Ethernet ports;
- for a frame designated for the device itself, accept and deliver the frame to the DLE without forwarding;
- for a frame designated for another device, accept the frame to its application layer and inspect both the destination and the source addresses. When the destination address of the frame is in the dynamic table, the GWD delivers the frame to the corresponding port in the dynamic table without forwarding to other ports. Otherwise, the GWD delivers the frame to all other ports. The GWD adds this entry to the dynamic table with source MAC address and port number information.

NOTE Dynamic table entries are automatically removed after the Ageing Time which is specified in IEEE 802.1D.

Figure 2 shows different structures of ND and GWD. In GWD, external Ethernet connection is connected to RRP ring network through MAC\_E and PHY\_E.

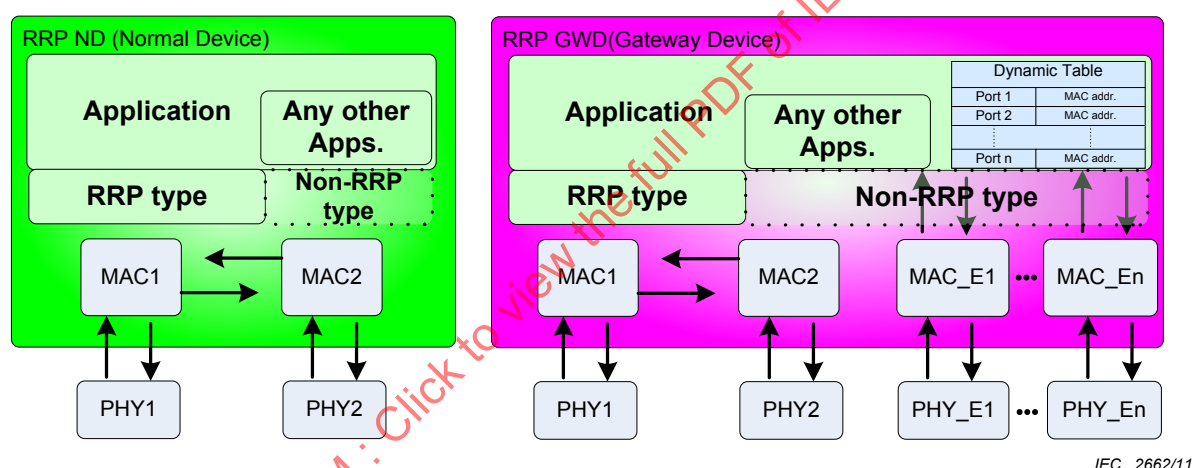


Figure 2 – Structures of ND and GWD

#### 4.2.3 Behaviours of the General Device (GD)

When an Ethernet frame is received at the MAC layer through the PHY, a RRP general device other than the ring network manager (RNM) or the line network manager (LNM), handles the received frame by taking one of the following actions, depending on the destination MAC address and the device address in the received frame:

- for a broadcast or multicast frame, accept and deliver the frame to the DLE, and forward the frame to the other port;
- for a frame designated for the device itself, accept and deliver the frame to the DLE without forwarding;
- for a frame designated for another device, do not accept the received frame, but forward the frame to the other port.

This frame forwarding procedure is processed by the internal hardware switch, so that it has little impact on the performance of the RRP protocol.



### 4.3 Link status monitoring

The RRP manages the network dynamically. When a link between two devices is established or released, it is automatically detected in the physical layer, as specified in ISO/IEC 8802-3:2000, Clause 24. This link status information is distributed and shared with every device on the network using NCMT messages, so that the network topology can be managed dynamically. The link status information is either “PHY\_LINK\_UP” or “PHY\_LINK\_DOWN” and the link status detection process is initiated by the sublayer of PHY service. A status of “PHY\_LINK\_UP” means that a RRP communication link is connected between two devices and it is possible to send frames through the link. A status of “PHY\_LINK\_DOWN” means that a RRP communication link is not established through an Ethernet MAC port and it is not possible to send frames through the port. By sharing all the link information on the network, all RRP devices on the network can determine the online network connectivity status. Figure 5 shows the intrinsic link status monitoring procedure of the RRP device.

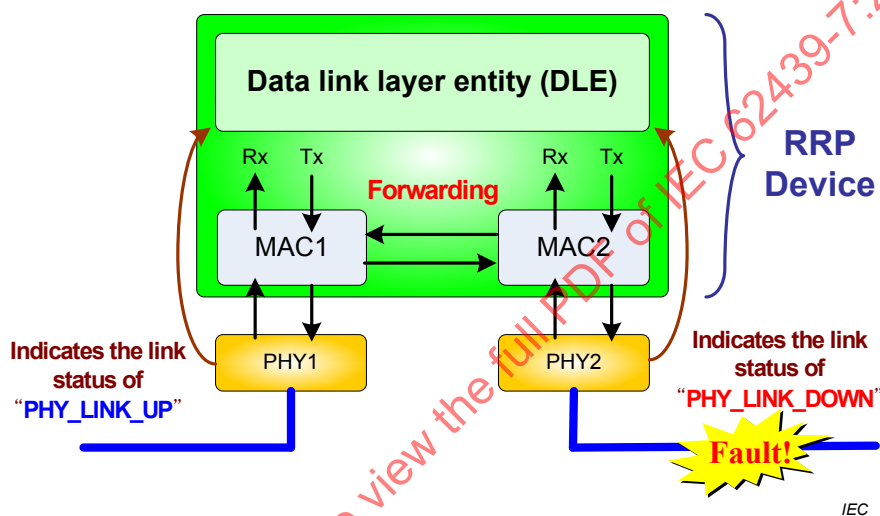


Figure 5 – Link status information

### 4.4 Error detection

A RRP device examines both frame validation and physical link status. Frame validation is examined using the frame check sequence (FCS) of ISO/IEC 8802-3:2000, Clause 3. The physical link status can be validated by a PHY link monitoring function. RRP uses a service of PHY sublayer to monitor link status.

### 4.5 Plug and play

When a new device joins an existing network, the new link information is broadcast via a NCMT message to every device on the network. The new device also collects existing link information from each device so that it can communicate to the other nodes on the network without manual configuration.

### 4.6 Network management information base (NMIB) management

A RRP device automatically manages network information and a path table. Network information and the path table are stored in the device's NMIB. All RRP devices in a network share link information via NCMT messages. Every device updates its network information and path table when it receives a NCMT message containing network information. Every device on the same network shares and gathers link information on the network to update its own network information and path table. Every device updates its network information and path table when it receives link status change information.

#### 4.7 Network recovery

When link failure or device failure is detected in a RRP ring network, the topology changes and the link status information is automatically broadcast to every device on the network. After broadcasting topology change information, every device on a network starts to update its own path table and tries to find new paths to other devices on the network. This process is operated in protocol machine and changing the blocking point of the network. Thus, devices can transmit messages to their destinations, while they are updating their NMIB.

#### 4.8 Automatic network configuration

RRP supports automatic network configuration. When the network topology changes, the RRP protocol machine of every device shares the changed network information, and then every device updates its own NMIB. RNMs or LNMs are automatically selected on the network according to the device UID and connection status.

When a device joins to a RRP network, the device broadcasts a NCMT message with its network information. Other devices in a RRP network receive the NCMT message then update their NMIB and reply to a new device. Thus NMIB is updated automatically through a joining process.

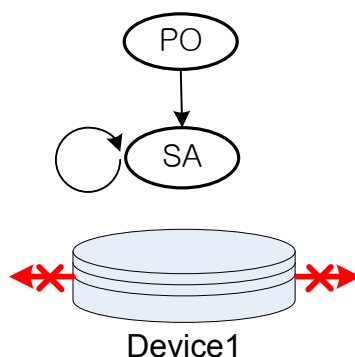
Similarly when a device detects any fault conditions, the device broadcasts a NCMT message to other devices in the network. So that network information is managed automatically by NCMT messages.

#### 4.9 RRP basic operating principle

The RRP network is established by following steps:

- power On (Initialization);
- establishment of line network;
- extension of line network;
- establishment of ring network;
- change topology from ring network to line network.

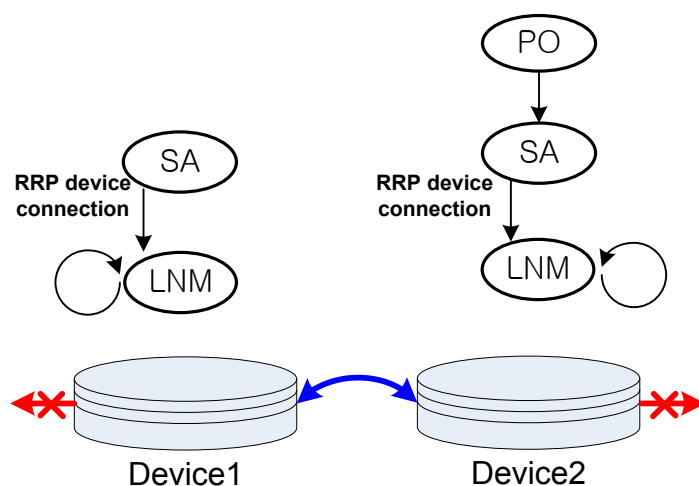
In the initialization phase, a RRP device becomes a stand-alone device. The device tries to find valid RRP connections on its ports. Figure 6 shows that Device1 is initialized and remained stand-alone device. In this phase, the device is in SA state.



IEC 2666/11

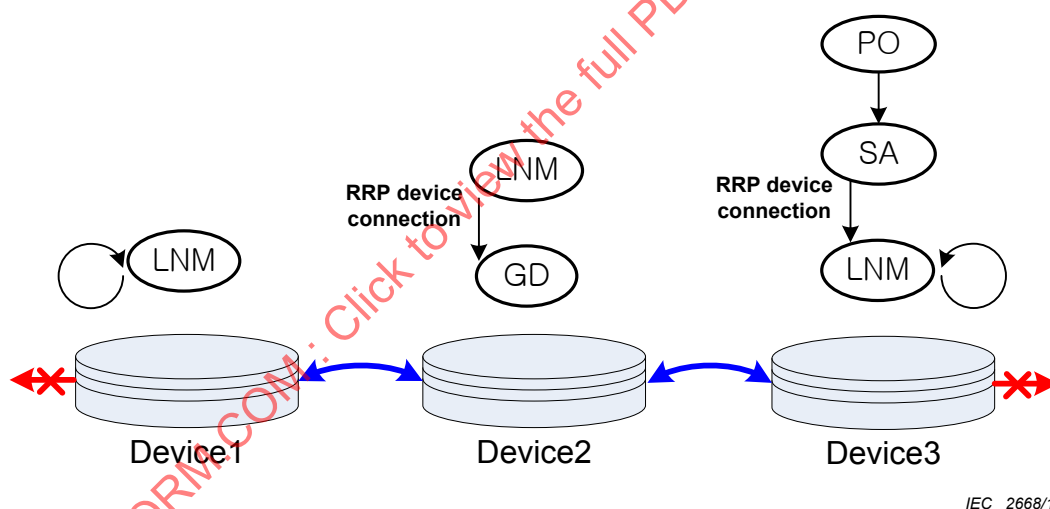
**Figure 6 – A device operation in initialization phase**

When the device detects a valid RRP connection on any port, it changes its state from SA to LNM and broadcast a NCMT message to the network. Figure 7 shows the device connection and its state.



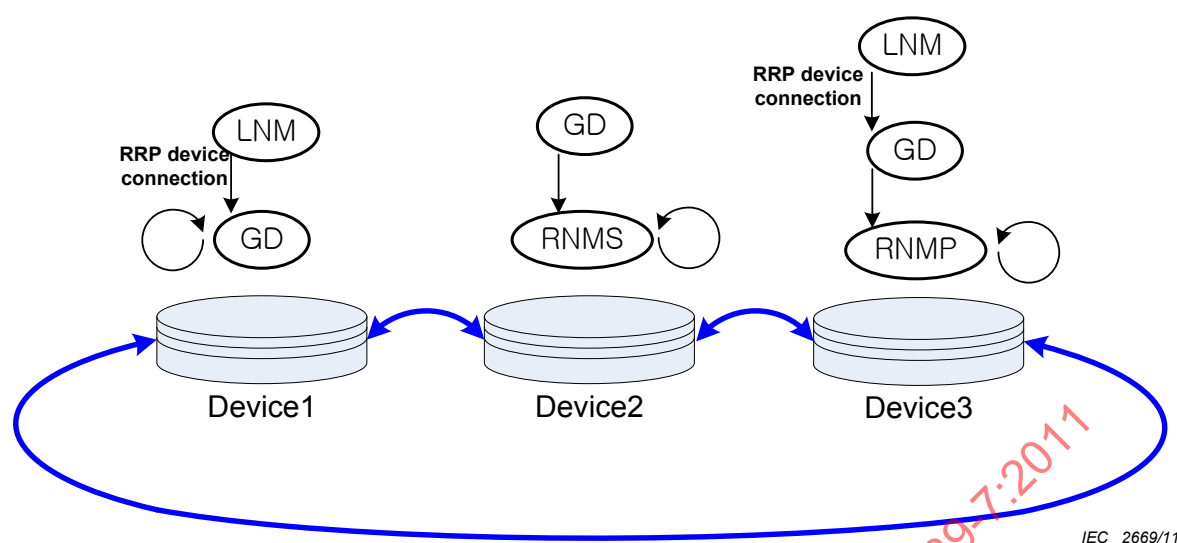
**Figure 7 – Devices operation in line network establishing phase**

If one or more devices added to the line network, the network is extended and two devices at both ends of the line network remain LNMs. This process is operated by RRP protocol machine in each device with NMIB. Figure 8 shows extension of line network operation.



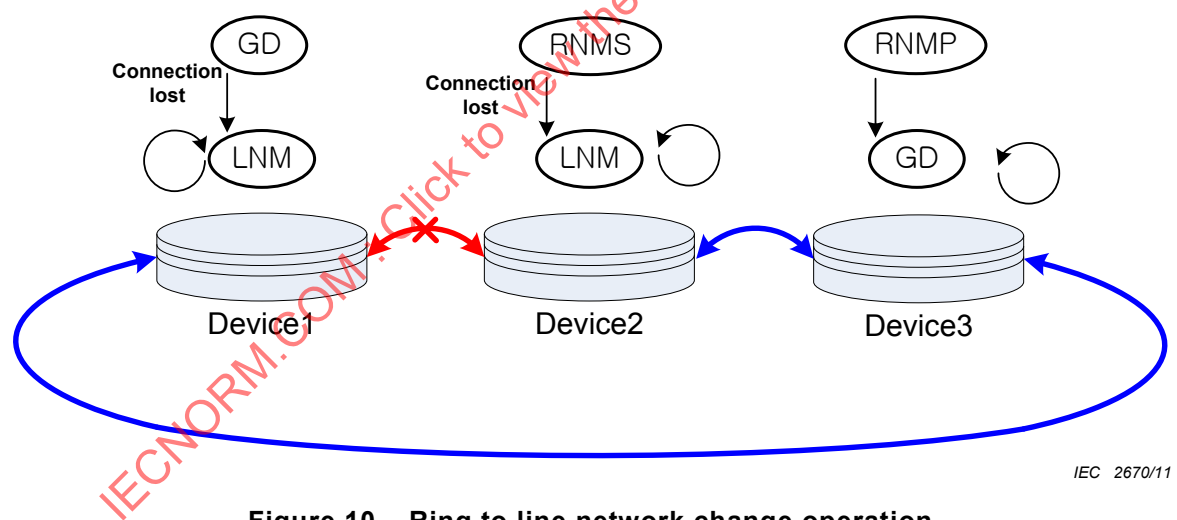
**Figure 8 – Extension of line network operation**

When both ends of the line network are connected, the line network is changing to ring network. Then the highest UID device sends a NCMT message as a RNMP. RNMP chooses R-port1 side neighbouring device as RNMS. Figure 9 shows a ring network establishment operation.



**Figure 9 – Ring network establishment operation**

If any fault condition is detected, neighbouring devices of the fault point broadcast NCMT message and change their state to LNM. Then the ring network is changed to line network. Figure 10 shows topology changing operation from ring network to line network. If the loss of valid connection is recovered, the network will be changed to ring network automatically as shown Figure 9.

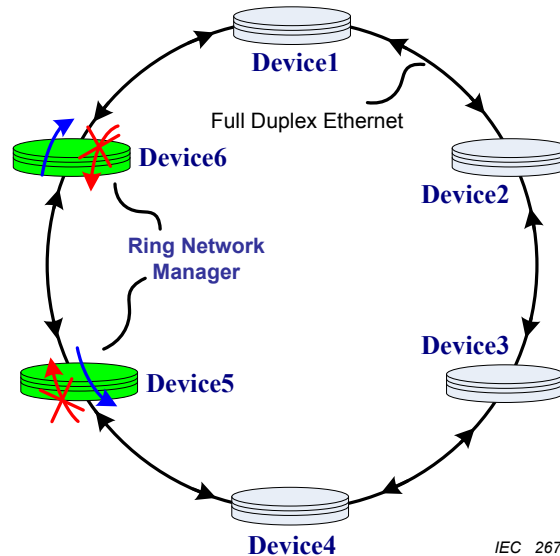


**Figure 10 – Ring to line network change operation**

## 5 RRP redundancy behaviours

### 5.1 Network topology

Figure 11 shows a basic example of a RRP ring network.



**Figure 11 – Ring topology**

When a new link is established between two end nodes in a line network, the network is automatically reconfigured as a ring network. Two RNMs are automatically selected to prevent the infinite circulation of any frame in a ring network. However, when a link failure is detected in the ring network, the ring network managers (RNMs) change their internal state to convert network topology from a ring to a line network. RNM could send NCMT message periodically to other RNM through the network to examine the network integrity.

## **5.2 Network recovery in ring network**

### **5.2.1 General**

The RRP network recovery process is simple and is handled within the protocol machine to provide fast recovery time upon network topology changes. When a link failure is detected in a ring network, the network is reconfigured automatically as a line network. A topology change from a ring to a line network occurs in the following three cases:

- link fault with the neighbouring device;
- link fault of the remote device;
- local device fault.

If a link failure occurs, the two devices nearest the fault point enter the LNM state and broadcast the NCMT message NCM\_LINE\_START. When a device receives NCM\_LINE\_START, it checks its connection status and changes its state to GD or LNM.

A cable break or power failure is treated in the same way as a link fault. A device fault also takes the same time to recover. Redundancy recovery time is determined from the time when a link fault occurs to the time when the path is recovered in a new direction.

The RRP specifies recovery times for the transition from a ring to a line network. Table 1 shows the parameter set for worst case RRP recovery times when the network parameter is set as Table 2.

**Table 1 – RRP network recovery parameter**

| Parameter             | 100BASE-X | 1000BASE-X | Meaning                                                                                                                                                                                                                                              |
|-----------------------|-----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T_fault_sense         | 350 us    | 2 ms       | the network fault sense time in PHY specified in ISO/IEC 8802-3:2000, Clause 24, Clause 25, Clause 26 and Clause 36<br>100BASE-TX: ANSI X3.263-1995, Clause 10<br>100BASE-FX: ISO/IEC 9314-3:1990, Clause 9<br>1000BASE-X: ANSI X3.230-1994, Annex I |
| T_state_transient     | 1 ms      | 1 ms       | the time spent switching the protocol machine from RNM or GD to LNM, in the RRP device                                                                                                                                                               |
| T_message_propagation | 6 ms      | 700 µs     | the message delivery time of NCM_LINE_START from a new LNM to the farthest device in a segment                                                                                                                                                       |
| T_recovery (max.)     | ≤ 8 ms    | ≤ 4 ms     | the total network recovery time in ms                                                                                                                                                                                                                |

**Table 2 – Parameters for calculation**

| Parameter        | Value      |            | Meaning                                                   |
|------------------|------------|------------|-----------------------------------------------------------|
|                  | 100BASE-X  | 1000BASE-X |                                                           |
| N                | 50         | 50         | the number of nodes between sending and receiving devices |
| T <sub>PKT</sub> | 24 µs      | 2,4 µs     | the packet transmit time                                  |
| NCMsize          | 112 octets | 112 octets | the message length of NCM_LINE_START message              |
| POsize           | 40 octets  | 40 octets  | the size of protocol overhead                             |
| LDR              | 100        | 1 000      | the link speed in Mbit/s                                  |
| T <sub>CPD</sub> | 0,5 µs     | 0,05 µs    | the cable propagation time of node for 100 m              |
| T <sub>SND</sub> | 50 µs      | 50 µs      | the sender stack traversal time including PHY and MAC     |
| T <sub>RCV</sub> | 50 µs      | 50 µs      | the receiver stack traversal time including PHY and MAC   |
| T <sub>NLD</sub> | 120 µs     | 12 µs      | the node latency delay time for worst case                |
|                  | 3 µs       | 0,3 µs     | the node latency delay time for best case                 |

The network recovery time can be calculated by Equation (1)

$$T_{\text{RECOVER}} = T_{\text{FS}} + T_{\text{RtoL}} + T_{\text{MSG}} \quad (1)$$

where

$T_{\text{RECOVER}}$  is the network recovery time in ms;

$T_{\text{FS}}$  is the link fault sense time in ms;

$T_{\text{RtoL}}$  is the device state transition delay time in µs;

$T_{\text{MSG}}$  is the message propagation time to the farthest device of NCM\_LINE\_START message in ms, see Equation (2).

The NCMT message, NCM\_LINE\_START, propagation time to the farthest device  $T_{\text{MSG}}$  can be calculated by Equation (2).

$$T_{\text{MSG}} = T_{\text{SND}} + T_{\text{PKT}} + T_{\text{CPD}} + \sum_{i=0}^N T_{\text{NLD}_i} + T_{\text{RCV}} \quad (2)$$

where

|              |                                                                         |
|--------------|-------------------------------------------------------------------------|
| $T_{MSG}$    | is the message propagation time in $\mu s$ ;                            |
| $T_{SND}$    | is the sender stack traversal time including PHY and MAC in $\mu s$ ;   |
| $T_{PKT}$    | is the packet transmit time in microseconds, see Equation (3);          |
| $T_{CPD}$    | is the cable propagation time for a node, in $\mu s$ ;                  |
| $T_{NLD\_i}$ | is the node latency time for node i in micro seconds, see Equation (4); |
| $T_{RCV}$    | is the receiver stack traversal time including PHY and MAC in $\mu s$ ; |
| $N$          | is the number of nodes between sending and receiving devices.           |

The packet transmit time  $T_{PKT}$  can be calculated by Equation (3)

$$T_{PKT} = \frac{(NCMsize + POsize) \times 8}{LDR} \quad (3)$$

where

|           |                                                         |
|-----------|---------------------------------------------------------|
| $T_{PKT}$ | is the packet transmit time on the medium, in $\mu s$ ; |
| $NCMsize$ | is the size of the NCM data unit in octets;             |
| $LDR$     | is the link data rate in bits per second;               |
| $POsize$  | is the size of the protocol overhead in octets.         |

The node latency time of node i  $T_{NLD\_i}$  can be calculated by Equation (4)

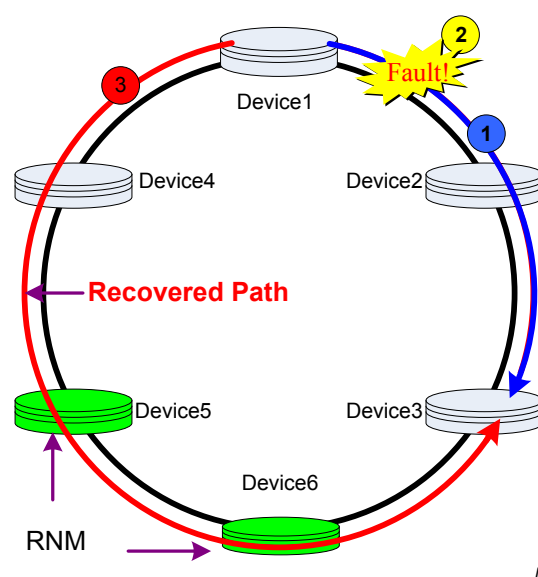
$$T_{NLD\_i} = T_{NPD\_i} + T_{PKT\_i} + \sum_{j=0}^M T_{TX\_PKT\_ij} \quad (4)$$

where

|                   |                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------|
| $T_{NLD\_i}$      | is the node latency delay time for node i in $\mu s$ ;                                                           |
| $T_{NPD\_i}$      | is the node propagation delay time for node i in $\mu s$ ;                                                       |
| $T_{PKT\_i}$      | is the packet transmit time for node i in $\mu s$ ;                                                              |
| $T_{TX\_PKT\_ij}$ | is the packet transmit time for packet j in ms within the port transmit queue of node i in front of this packet; |
| $M$               | is the number of packets in the port transmit queue of node i in front of this packet.                           |

## 5.2.2 Link fault between neighbouring devices

Figure 12 shows an example of a link fault between neighbouring devices in a ring network.



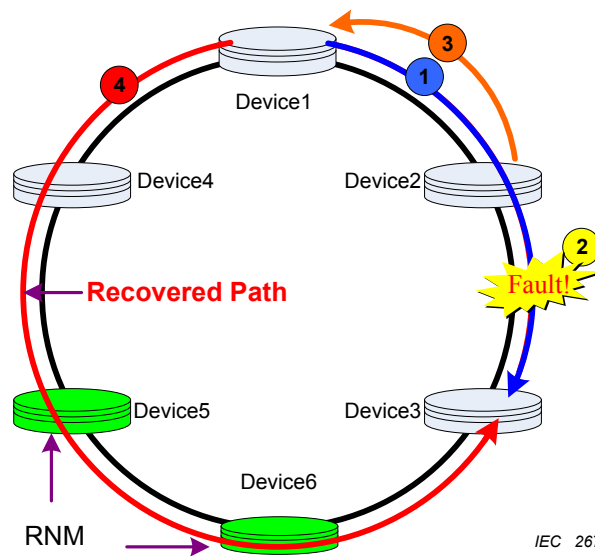
**Figure 12 – Link fault between neighbouring devices**

If the link between Device1 and Device2 is disconnected when Device1 tries to send a frame to Device3, the link fault event is triggered spontaneously by a hardware signal, and it is detected by the RRP protocol machine in Device1. Device1 then decides that the link is not available for data transmission, and so the ring network should be reconfigured as a line network, by changing the state of Device1 to LNM. Device1 broadcasts a NCM\_LINE\_START message and modifies the destination R-port to Device3. Device1 transmits the frame to the new destination R-port. Device2 also broadcasts a NCM\_LINE\_START message, to announce the link fault, and changes its own state to LNM.

In case that Device2 could not detect the link fault, Device5 receives NCM\_LINE\_START message from Device1 and change its state from RNM to GD. Then, Device5 relays NCM\_LINE\_START message to Device6. Thus, other devices of the network recognize topology changing.

### 5.2.3 Link fault of remote device

Figure 13 shows an example of a link fault of the remote device in a ring network.

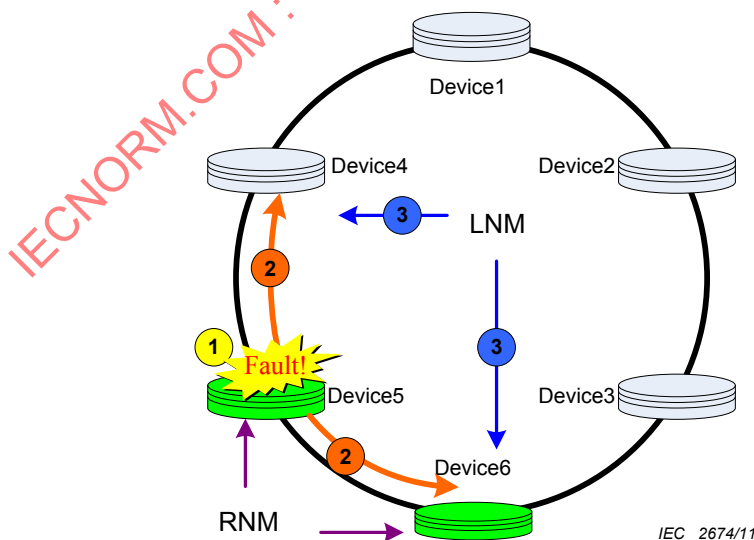


**Figure 13 – Link fault of remote device**

If the link between Device2 and Device3 is disconnected when Device1 tries to send a frame to Device3, the link fault event is triggered spontaneously by the hardware signal and is detected by the RRP protocol machine in Device2. At this point, Device2 broadcasts an NCM\_LINE\_START message indicating that the link is not available and that the ring network should be reconfigured as a line network. Device1 modifies the destination R-port to Device3 as the other R-port and transmits the frame through the new R-port. Device3 also broadcasts a NCM\_LINE\_START message, to announce the link fault, and changes its own state to LNM.

#### 5.2.4 Device fault on a RNM

Figure 14 shows an example of a device fault on a RNM in a ring network.



**Figure 14 – Device fault on a RNM**

If a fault occurs on Device5, Device6 and Device4 sense link failure on their respective Device5 side R-ports. Device6 and Device4 both broadcast NCM\_LINE\_START messages to

other devices and change their state of to LNM. A GD fault is treated in same way as a RNM fault.

### 5.3 Automatic Ring Network Manager (RNM) election procedure

#### 5.3.1 General

To prevent infinite frame circulation in a RRP ring network, the primary RNM (RNMP) and the secondary RNM (RNMS) are selected automatically using their device unique IDs (UID), via the following steps:

- when two LNMs are connected, each LNM sends NCMT message as multicast or broadcast;
- every device which receives NCMT message responds with its UID as multicast or broadcast;
- when a device receives its own responded NCMT message, the device detects the network is a ring network and does not forward the message;
- the device with the highest device UID value in the ring network is selected automatically as the RNMP;
- the RNMP sends an NCM\_RING\_START message including the RNMS assignment request to the neighbouring device connected through its both of R-port1 and R-port2;
- the RNMS replies with an NCM\_ACK\_RNMS message to the RNMP;
- automatic ring network configuration is completed with RNMP and RNMS;
- the RNMP blocks frame forwarding to RNMS side R-port and RNMS also blocks frame forwarding to RNMP side R-port to prevent looping on the ring.

#### 5.3.2 Primary RNM (RNMP)

A RRP device realizes that the network is configured as a ring topology when the NCMT message generated by the device itself is received through the other port. In a ring network, the device with the highest device UID value is selected as the RNMP. When a RRP device detects that the network is configured as a ring network, each device tries to find the device in the path table that has the highest device UID value. Thus, competition to select the RNMP is not necessary in RRP. If a remote device has the highest UID on the network, the other devices wait for the NCM\_RING\_START message from it. If a local device has the highest UID on the network, then it is selected as the RNMP, the RRP protocol machine disables both frame forwarding functions to prevent looping on the network, and generates an NCM\_RING\_START message, including the RNMS information of the device connected through both of R-port1 and R-port2 of the RNMP.

When the RNMP receives the NCM\_ACK\_RNMS message from the RNMS, it disables the frame forwarding function in the RNMS direction, but keeps the opposite direction enabled. The RNMS disables the frame forwarding function in the RNMP direction and enables it in the opposite direction.

RNMP could send NCM\_CHECK\_RNMS message to RNMS as a network integrity check frame periodically. If the response of NCM\_CHECK\_RNMS message is not arrived within the timer RRP\_ChkRNMST from RNMS and exceed the number of network integrity retry count, the RNMP figures out the network has an error.

NOTE 1 The device UID has a unique value on the network. The RNMP and RNMS are therefore selected automatically even in the case of a device address collision (See 5.5).

NOTE 2 NCM\_CHECK\_RNMS and NCM\_ACK\_RNMS messages are using to choose RNMS and to confirm RNMS by RNMP. But RRP supports these messages could be used to check network integrity by user. The timer and the retry count are specified as local variables (See Table 32).

### 5.3.3 Secondary RNM (RNMS)

The RNMS is assigned by the RNMP. When a GD device receives an NCM\_RING\_START message from the RNMP, the RRP protocol machine compares the local device UID to the RNMS device UID in the received NCM\_RING\_START message. If the device is not designated as the RNMS, the RRP protocol machine enables both frame forwarding functions in the GD state device. If the device is designated as the RNMS, the RRP protocol machine enters the RNMS state and transmits NCM\_ACK\_RNMS to the RNMP through the received R-port of the NCM\_RING\_START message and NCM\_CHECK\_RNMS message from the RNMP. Additionally, the RNMS device disables the frame forwarding function in the RNMP direction, but enables it in the opposite direction.

## 5.4 Path management

### 5.4.1 General

The RRP device provides path information about each RRP device on the network. Path management is calculated from the hop count. The hop count indicates how many frame forward operations are required to transfer a frame to the destination device.

### 5.4.2 Path in a line topology network

In a line network, only one path is possible to the destination device. Figure 15 shows an example of path management in a line topology network.

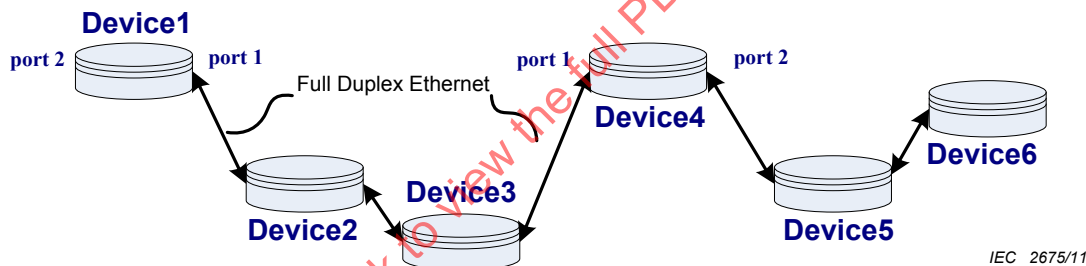


Figure 15 – Path management in a line topology network

Table 3 shows the path table of Device1 and Table 4 shows the path table of Device4 in Figure 15. In a line network, only one path is possible between any two devices.

Table 3 – Path table of Device1 in a line topology network

| R-port           | Destination |         |         |         |         |
|------------------|-------------|---------|---------|---------|---------|
|                  | Device2     | Device3 | Device4 | Device5 | Device6 |
| R-port1          | 0 hop       | 1 hop   | 2 hops  | 3 hops  | 4 hops  |
| R-port2          | Invalid     | Invalid | Invalid | Invalid | Invalid |
| Preferred Port   | R-port1     | R-port1 | R-port1 | R-port1 | R-port1 |
| Destination Port | R-port1     | R-port1 | R-port1 | R-port1 | R-port1 |

**Table 4 – Path table of Device4 in a line topology network**

| R-port           | Destination |         |         |         |         |
|------------------|-------------|---------|---------|---------|---------|
|                  | Device1     | Device2 | Device3 | Device5 | Device6 |
| R-port1          | 2 hops      | 1 hop   | 0 hop   | Invalid | Invalid |
| R-port2          | Invalid     | Invalid | Invalid | 0 hop   | 1 hop   |
| Preferred Port   | R-port1     | R-port1 | R-port1 | R-port2 | R-port2 |
| Destination Port | R-port1     | R-port1 | R-port1 | R-port2 | R-port2 |

### 5.4.3 Path in a ring topology network

In a ring network, two paths are possible between any two devices: the clockwise path and the counter clockwise path. However, a frame cannot be forwarded across the RNMP or RNMS. Therefore, it is impossible to transfer a frame using the path including the RNMP or RNMS. Figure 16 shows an example of path management in a ring network.

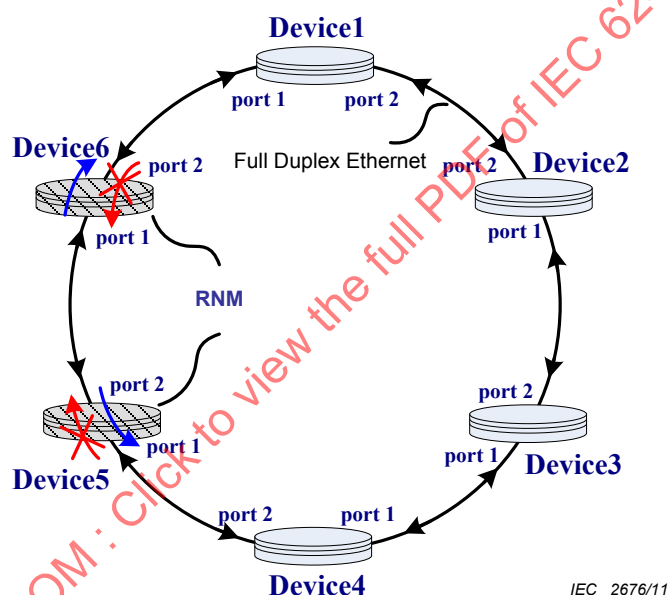
**Figure 16 – Path management in a ring topology network**

Table 5 shows the path table of Device1 in Figure 16. The shortest path from Device1 to Device5 is in the R-port1 direction, but this path is blocked by the RNM, Device6. In this case, the destination path is determined to be in the R-port2 direction.

**Table 5 – Path table of Device1 in a ring topology network**

| R-port                                                                                                                    | Destination |         |               |                                                      |               |
|---------------------------------------------------------------------------------------------------------------------------|-------------|---------|---------------|------------------------------------------------------|---------------|
|                                                                                                                           | Device2     | Device3 | Device4       | Device5 (RNM)                                        | Device6 (RNM) |
| R-port1                                                                                                                   | 4 hops      | 3 hops  | 2 hops        | 1 hop                                                | 0 hop         |
| R-port2                                                                                                                   | 0 hop       | 1 hop   | 2 hops        | 3 hops                                               | 4 hops        |
| Preferred Port                                                                                                            | R-port2     | R-port2 | Don't care    | R-port1<br>(R-port1 direction is blocked by Device6) | R-port1       |
| Destination Port                                                                                                          | R-port2     | R-port2 | R-port1(NOTE) | R-port2                                              | R-port1       |
| NOTE If both paths have the same hop counts and they are not blocked by the RNMs, the R-port1 direction is always chosen. |             |         |               |                                                      |               |

Table 6 shows the path table of Device3 in Figure 16.

**Table 6 – Path table of Device3 in a ring topology network**

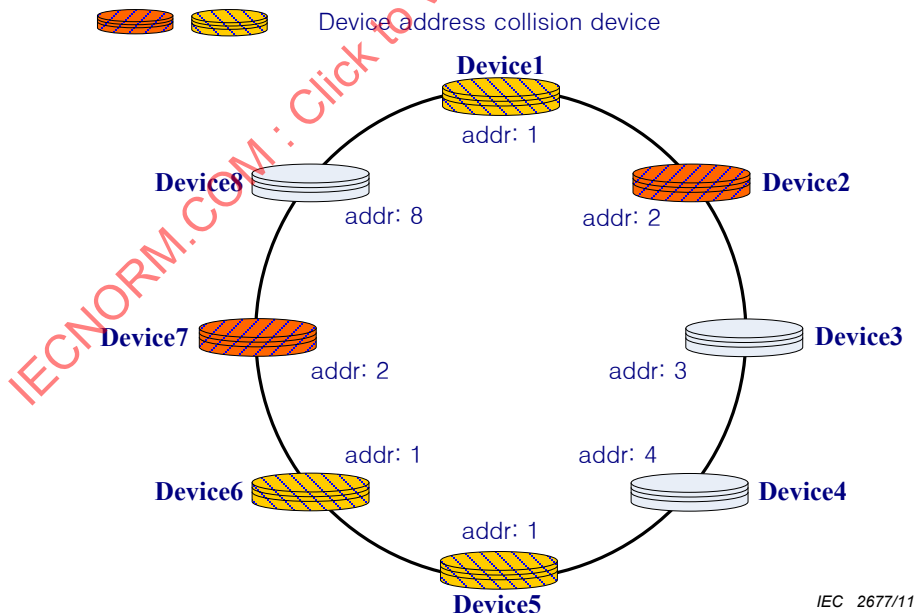
| R-port           | Destination |         |         |               |                                                         |
|------------------|-------------|---------|---------|---------------|---------------------------------------------------------|
|                  | Device1     | Device2 | Device4 | Device5 (RNM) | Device6 (RNM)                                           |
| R-port1          | 3 hops      | 4 hops  | 0 hop   | 1 hop         | 2 hops                                                  |
| R-port2          | 1 hop       | 0 hop   | 4 hops  | 3 hops        | 2 hops                                                  |
| Preferred Port   | R-port2     | R-port2 | R-port1 | R-port1       | Don't care<br>(R-port1 direction is blocked by Device5) |
| Destination Port | R-port2     | R-port2 | R-port1 | R-port1       | R-port2                                                 |

### 5.5 Device address collision

A RRP device address is configured manually by hardware settings (e.g., rotary switch) or set by software (e.g., portable terminal). The device address may be duplicated in other devices on the network because of a configuration error. When this happens, the device is unable to communicate with other devices. This type of device address collision event on the network is detected automatically by the RRP protocol machine.

RRP uses 8 octets UID instead of device address to manage network resources. UID is made combination of device address and MAC address so that UID is a unique value in the network. Network and UID information is stored in NMIB of each device. Thus, devices of network could know which device has duplicated address.

Figure 17 shows an example of device address collision in a ring network.



**Figure 17 – RRP device address collision in a ring network**

Devices1, 5, and 6 have the same device address value of 1. Device2 and Device7 have the same device address value of 2. Therefore, three device address collision events are detected by the RRP protocol machine. Device address collision events are counted by the collision counters in the network management information base and shared with every device on the network.

Table 7 shows the device address collision information for the situations in Figure 17 representing the device address collision detection mechanism using the device UUIDs.

**Table 7 – Device address collision information**

| Device name | Device address | MAC address    | Device UUID        | Collision |
|-------------|----------------|----------------|--------------------|-----------|
| Device1     | 0x0001         | 0x002233445511 | 0x0001002233445511 | Collision |
| Device2     | 0x0002         | 0x002233445522 | 0x0002002233445522 | Collision |
| Device3     | 0x0003         | 0x002233445533 | 0x0003002233445533 | —         |
| Device4     | 0x0004         | 0x002233445544 | 0x0004002233445544 | —         |
| Device5     | 0x0001         | 0x002233445555 | 0x0001002233445555 | Collision |
| Device6     | 0x0001         | 0x002233445566 | 0x0001002233445566 | Collision |
| Device7     | 0x0002         | 0x002233445577 | 0x0002002233445577 | Collision |
| Device8     | 0x0008         | 0x002233445588 | 0x0008002233445588 | —         |

The device address is configured manually by the operator to be some value in the range 0-255. The MAC address is a 6 octet unique ISO/IEC 8802-3:2000 Ethernet MAC address. The RRP device UUID contains 2 octet RRP device address and the 6 octet MAC address. Therefore, the RRP device UUID has a unique 8 octet value on the network that cannot be duplicated. The RRP device UUID is used to recognize a specific device on the network.

A RRP device address collision is detected by the RRP protocol machine when devices with different device UUIDs are configured with the same device address. If a local device address collision is detected, the device sets the device address collision flag in the local device flags and generates a device address collision event. If a remote device address collision is detected, the device sets the device address collision flag in the network flags and generates a network device address collision event.

## 6 RRP class specification

### 6.1 General

The RRP Application Service Element (ASE) defines one object type.

### 6.2 Template

A RRP object is described by the following template:

**ASE:** Ring-based redundancy ASE  
**CLASS:** Ring-based redundancy  
**CLASS ID:** not used  
**PARENT CLASS:** IEEE 802.3 Ring-based Redundancy Protocol

#### ATTRIBUTES:

|   |     |                |                                  |
|---|-----|----------------|----------------------------------|
| 1 | (m) | Key Attribute: | Local device address             |
| 2 | (m) | Attribute:     | Local device flags               |
| 3 | (m) | Attribute:     | Local device state               |
| 4 | (m) | Attribute:     | Local device unique ID           |
| 5 | (m) | Attribute:     | Device UUID for R-port1          |
| 6 | (m) | Attribute:     | Device UUID for R-port2          |
| 7 | (m) | Attribute:     | Local device MAC address         |
| 8 | (m) | Attribute:     | Local device R-port1 information |

|       |     |             |                                              |
|-------|-----|-------------|----------------------------------------------|
| 9     | (m) | Attribute:  | Local device R-port2 information             |
| 10    | (m) | Attribute:  | Local device protocol version                |
| 11    | (m) | Attribute:  | Local device type                            |
| 12    | (m) | Attribute:  | Local device description                     |
| 13    | (m) | Attribute:  | Hop count                                    |
| 14    | (m) | Attribute:  | FamilyRes frame waiting time                 |
| 15    | (m) | Attribute:  | AdvThis frame waiting time                   |
| 16    | (m) | Attribute:  | AckRNMS frame waiting time                   |
| 17    | (m) | Attribute:  | Ring state change timeout                    |
| 18    | (m) | Attribute:  | Diagnostic information (Network, Path Table) |
| 19    | (c) | Constraint: | Diagnostic information = Network             |
| 19.1  | (m) | Attribute:  | Network topology                             |
| 19.2  | (m) | Attribute:  | Collision count                              |
| 19.3  | (m) | Attribute:  | Device count                                 |
| 19.4  | (m) | Attribute:  | Topology change count                        |
| 19.5  | (m) | Attribute:  | Last topology change time                    |
| 19.6  | (m) | Attribute:  | RNMP device UID                              |
| 19.7  | (m) | Attribute:  | RNMS device UID                              |
| 19.8  | (m) | Attribute:  | LNMP device UID for R-port1                  |
| 19.9  | (m) | Attribute:  | LNMP device UID for R-port2                  |
| 19.10 | (m) | Attribute:  | Network flags                                |
| 20    | (c) | Constraint: | Diagnostic information = Path Table          |
| 20.1  | (m) | Attribute:  | Peer device address                          |
| 20.2  | (m) | Attribute:  | Peer device hop count for R-port1            |
| 20.3  | (m) | Attribute:  | Peer device hop count for R-port2            |
| 20.4  | (m) | Attribute:  | Preferred R-port for peer device             |
| 20.5  | (m) | Attribute:  | Destination R-port for peer device           |
| 20.6  | (m) | Attribute:  | Peer device state                            |
| 20.7  | (m) | Attribute:  | Peer device MAC address                      |
| 20.8  | (m) | Attribute:  | Peer device R-port1 information              |
| 20.9  | (m) | Attribute:  | Peer device R-port2 information              |
| 20.10 | (m) | Attribute:  | Peer device protocol version                 |
| 20.11 | (m) | Attribute:  | Peer device type                             |
| 20.12 | (m) | Attribute:  | Peer device description                      |
| 20.13 | (m) | Attribute:  | Peer device UID                              |
| 20.14 | (m) | Attribute:  | Device UID for R-port1 of peer device        |
| 20.15 | (m) | Attribute:  | Device UID for R-port2 of peer device        |
| 20.16 | (m) | Attribute:  | In net count of peer device                  |
| 20.17 | (m) | Attribute:  | In net time of peer device                   |
| 20.18 | (m) | Attribute:  | Out net count of peer device                 |
| 20.19 | (m) | Attribute:  | Out net time of peer device                  |

#### SERVICES:

- 1 (m) OpsService: Set Device Information
- 2 (m) OpsService: Get Device Information
- 3 (m) OpsService: Get Network Information
- 4 (m) OpsService: Get Path Table Information

### 6.3 Attributes

NOTE 1 The data type of each attribute is following definition of the standard IEC 61158-4-21:2010, Clause 4.

#### Local device address

This key attribute defines the RRP local device address that designates a single RRP device on a specific local link; its value is constrained to the range 0-255.

Data type: Unsigned16

NOTE 2 The local device address may be provided by hardware settings (e.g., rotary switch) or set by software.

#### Local device flags

This attribute specifies the flags for events that occurred in a local device. The meaning for each bit is as follows:

- Bit 0: device address collision
- Bit 1: device state changed

Data type: Unsigned16

#### Local device state

This attribute specifies the local device state. The device state shall have one of the following values:

- 0 = invalid
- 1 = SA (standalone state)
- 2 = LNM (line network manager state)
- 3 = GD (general device state)
- 4 = RNMP (primary ring network manager state)
- 5 = RNMS (secondary ring network manager state)

Data type: Unsigned8

#### Local device MAC address

This attribute defines the 6 octet ISO/IEC 8802-3:2000 Ethernet MAC address of the local device. Because a RRP device has two Ethernet MAC ports, both MAC addresses should be identical.

Data type: Unsigned48

#### Local device unique ID

This attribute defines the unique 8 octet identification that identifies a RRP device in a network. It is a combination of the 6 octet ISO/IEC 8802-3:2000 MAC address and the 2 octet device address. The individual bits shall have the following meaning:

- Bit 0 – 15: device address
- Bit 16 – 63: ISO/IEC 8802-3:2000 MAC Address

Data type: UniqueDeviceID64

#### Device UID for R-port1

This attribute defines the UID of the device that is linked through the R-port1. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

#### Device UID for R-port2

This attribute defines the UID of the device that is linked through the R-port2. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

#### **Local device R-port1 information**

This attribute defines the port information for R-port1. The meaning for each bit is as follows:

- Bit 0: R-port link down
- Bit 1: received Family\_Frame.Cnf from R-port
- Bit 2: waiting for AdvThis\_Frame.Cnf from R-port
- Bit 3: waiting for MediaLinked\_Frame.Ind from R-port
- Bit 4: device state confirm

Data type: Unsigned8

#### **Local device R-port2 information**

This attribute defines the port information for R-port2. It shall have one of the values defined for local device R-port1 information.

Data type: Unsigned8

#### **Local device protocol version**

This attribute defines the RRP protocol version of the local device. The individual bits shall have the following meaning:

- Bit 0 – 1: major version
- Bit 2 – 4: minor version
- Bit 5 – 7: reserved

Data type: Unsigned8

#### **Local device type**

This attribute defines the local device type that represents the general function of the device. The individual bits shall have the following meaning:

- Bit 0 – 7: general device type
- Bit 8 – 15: application-specific device type

Data type: Unsigned16

#### **Local device description**

This attribute defines a description of the local device and contains any string defined by the user using the Set Device Information service.

Data type: VisibleString[16]

#### **Hop count**

This attribute defines the count of the number of devices between two devices. When the RRP device receives the NCM frame, the RRP device saves the received hop count value in this variable, then increments the hop counts in the received frame by 1 and transmits the frame through the other R-port. In this way, each device builds its own path table with a hop count for R-port1 and a hop count for R-port2.

Data type: Unsigned16

#### **FamilyRes frame waiting time**

This attribute defines the time interval between sending the FamilyReq frame and receiving the FamilyRes frame. This attribute shall be configured by the user using the Set Device Information service.

Data type: Unsigned32

#### **AdvThis frame waiting time**

This attribute defines the time interval between sending the MediaLinked frame and receiving the AdvThis frame. This attribute shall be configured by the user using the Set Device Information service.

Data type: Unsigned32

**AckRNMS frame waiting time**

This attribute defines the time interval between sending the RingStart frame and receiving the AckRNMS frame. This attribute shall be configured by the user using the Set Device Information service.

Data type: Unsigned32

**Ring state change timeout**

This attribute defines the timeout to generate the event for changing the RNMP device state. This attribute shall be configured by the user using the Set Device Information service.

Data type: Unsigned32

**Diagnostic information**

This attribute defines the type of diagnostic information. The type of diagnostic information shall have the two values of Network and Path Table.

Data type: Unsigned8

**Network topology**

This attribute defines the type of network topology. It shall have one of the following values:

- 0 = invalid
- 1 = NET\_TPG\_SA (standalone)
- 2 = NET\_TPG\_LINE (line topology)
- 3 = NET\_TPG\_RING (ring topology)

Data type: Unsigned8

**Collision count**

This attribute defines the device address collision count for remote devices. Values are in the range 0-255.

Data type: Unsigned8

NOTE 3 The value is incremented by the RRP protocol machine when a remote device address collision is detected, and the value is decremented when the collision is cleared.

**Device count**

This attribute defines the total number of devices on the network. Values are in the range 1-255.

Data type: Unsigned16

**Topology change count**

This attribute defines the topology change count. The values are in the range 0-65 535.

Data type: Unsigned16

NOTE 4 The value is incremented by the RRP protocol machine when the network changes from ring to line or from line to ring topology.

**Last topology change time**

This attribute defines the date and time when the network topology was last changed.

Data type: TIMEOFDAY

**RNMP device UID**

This attribute defines the device UID selected as the RNMP on the network. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

### **RNMS device UID**

This attribute defines the UID of the device selected as the RNMS on the network. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

### **LNM device UID for R-port1**

This attribute defines the UID of the device selected as the LNM in the R-port1 direction. In a RRP line network, the two end devices are automatically selected as the LNMs. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

### **LNM device UID for R-port2**

This attribute defines the UID of the device selected as the LNM in the R-port2 direction. In a RRP line network, the two end devices are automatically selected as the LNMs. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

### **Network flags**

This attribute defines the flags for events that occurred in the network. The meaning for each bit is as follows:

- Bit 0: network topology has changed
- Bit 1: device address collision has been detected in the network
- Bit 2: new device has joined the network
- Bit 3: device has left the network

Data type: Unsigned16

NOTE 5 Each bit is set by the RRP protocol machine when the corresponding event occurs.

### **Peer device address**

This attribute defines the peer device address stored in the path table. The value is constrained to the range 0-255.

Data type: Unsigned16

### **Peer device hop count for R-port1**

This attribute defines the frame forwarding counts for sending a frame from the local device to the peer device through the R-port1.

Data type: Unsigned16

### **Peer device hop count for R-port2**

This attribute defines the frame forwarding counts for sending a frame from the local device to the peer device through the R-port2.

Data type: Unsigned16

### **Preferred R-port for peer device**

This attribute defines the preferred R-port for sending a frame from the local device to the peer device without regard for the RNMP or RNMS. It shall have one of the following values:

- 0 = invalid
- 1 = R-port1
- 2 = R-port2

Data type: Unsigned8

NOTE 6 It is defined as the R-port that has the smaller hop count value for the peer device. If the R-port1 and R-port2 hop counts have the same value, R-port1 is selected as the preferred R-port.

**Destination R-port for peer device**

This attribute defines the destination R-port for sending a frame from the local device to the peer device.

Data type: Unsigned8

NOTE 7 In a line network, this variable has the same value as the preferred R-port. However, in a ring network, this variable is determined based on the RNMP and RNMS positions, because the preferred path may be blocked by the RNMP or RNMS. In this case, the destination R-port is selected as the other R-port. It should have one of the values defined for the preferred R-port.

**Peer device state**

This attribute defines the peer device state stored in the path table. It shall have one of the values defined for the local device state.

Data type: Unsigned8

**Peer device MAC address**

This attribute defines the peer device 6 octets ISO/IEC 8802-3:2000 Ethernet MAC address stored in the path table.

Data type: Unsigned48

**Peer device R-port1 information**

This attribute defines the peer device R-port1 information stored in the path table. It shall have one of the values defined for the local device R-port1 information.

Data type: Unsigned8

**Peer device R-port2 information**

This attribute defines the peer device R-port2 information stored in the path table. It shall have one of the values defined for the local device R-port1 information.

Data type: Unsigned8

**Peer device protocol version**

This attribute defines the peer device RRP protocol version stored in the path table. The individual bits shall have the same meaning as the local device protocol version.

Data type: Unsigned8

**Peer device type**

This attribute defines the peer device application device type stored in the path table. The individual bits shall have the same meaning as the local device type.

Data type: Unsigned16

**Peer device description**

This attribute defines the peer device description stored in the path table.

Data type: VisibleString[16]

**Peer device UID**

This attribute defines the peer device unique ID stored in the path table. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

**Device UID for R-port1 of peer device**

This attribute defines the UID of the device that is linked through the R-port1 of the peer device and stored in the path table. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

### Device UID for R-port2 of peer device

This attribute defines the UID of device that is linked through the R-port2 of the peer device and stored in the path table. The individual bits shall have the same meaning as the local device unique ID.

Data type: UniqueDeviceID64

### In net count of peer device

This attribute defines the number of times that the peer device has joined the network

Data type: Unsigned16

NOTE 8 When a line network is merged into an existing network, the variables for the newly joined devices are incremented together.

### In net time of peer device

This attribute defines the date and time when the peer device last joined in the network.

Data type: TIMEOFDAY

### Out net count of peer device

This attribute defines the number of times that the peer device has been disconnected from the network. When a device or a group of devices is disconnected from the network, the variables for the disconnected devices are incremented together.

Data type: Unsigned16

### Out net time of peer device

This attribute defines the date and time when the device was last disconnected from the network.

Data type: TIMEOFDAY

## 7 RRP services specification

### 7.1 Set device information

This service is used to assign new values to the variables of the local device information.

The parameters of this service are specified in Table 8.

**Table 8 – Parameters of set device information service**

| Parameter name                | Req | Ind  | Rsp | Cnf |
|-------------------------------|-----|------|-----|-----|
| Argument                      | M   | M(=) |     |     |
| Service ID                    | M   | M(=) |     |     |
| Invoke ID                     | M   | M(=) |     |     |
| Local device address          | M   | M(=) |     |     |
| Local device MAC address      | M   | M(=) |     |     |
| Local device protocol version | M   | M(=) |     |     |
| Local device type             | M   | M(=) |     |     |
| Local device description      | M   | M(=) |     |     |
| FamilyRes frame waiting time  | M   | M(=) |     |     |
| AdvThis frame waiting time    | M   | M(=) |     |     |
| AckRNMS frame waiting time    | M   | M(=) |     |     |
| Ring state change timeout     | M   | M(=) |     |     |

| Parameter name | Req | Ind | Rsp | Cnf  |
|----------------|-----|-----|-----|------|
| Result(+)      |     |     | S   | S(=) |
| Service ID     |     |     | M   | M(=) |
| Invoke ID      |     |     | M   | M(=) |
| Status Code    |     |     | M   | M(=) |
| Result(-)      |     |     | S   | S(=) |
| Service ID     |     |     | M   | M(=) |
| Invoke ID      |     |     | M   | M(=) |
| Status Code    |     |     | M   | M(=) |

### Argument

The argument conveys the service specific parameters of the service request.

#### Service ID

This parameter contains information sufficient for local identification of the RRP device to be used to convey the service.

#### Invoke ID

This parameter identifies this invocation of the service.

#### Local device address

This parameter contains the value for the RRP device address.

#### Local device MAC address

This parameter contains the MAC address of the RRP device.

#### Local device protocol version

This parameter contains the protocol version of the RRP device.

#### Local device type

This parameter contains the RRP device type that represents the general function of the RRP device.

#### Local device description

This parameter contains a description of the RRP device.

#### FamilyRes frame waiting time

This parameter contains the value for the FamilyRes frame waiting time of the RRP device.

#### AdvThis frame waiting time

This parameter contains the value for the AdvThis frame waiting time of the RRP device.

#### AckRNMS frame waiting time

This parameter contains the value for the AckRNMS frame waiting time of the RRP device.

#### Ring state change timeout

This parameter contains the value for the ring state change timeout of the RRP device.

### Result(+)

This parameter indicates that the service request succeeded.

#### Service ID

This parameter contains information sufficient for local identification of the RRP device to be used to convey the service.

#### Invoke ID

This parameter identifies this invocation of the service.

#### Status Code

This parameter indicates whether the service was processed successfully. If an error occurred, it indicates the type of error.

Data type: Unsigned8

#### Result(-)

This parameter indicates that the service request failed.

#### Service ID

This parameter contains information sufficient for local identification of the RRP device to be used to convey the service.

#### Invoke ID

This parameter identifies this invocation of the service.

#### Status Code

This parameter indicates whether the service was processed successfully. If an error occurred, it indicates the type of error.

Data type: Unsigned8

### 7.2 Get device information

This service is used to obtain the local device information from the RRP device.

The parameters of this service are specified in Table 9.

**Table 9 – Parameters of get device information service**

| Parameter name                   | Req | Ind  | Rsp | Cnf  |
|----------------------------------|-----|------|-----|------|
| Argument                         | M   | M(=) |     |      |
| Service ID                       | M   | M(=) |     |      |
| Invoke ID                        | M   | M(=) |     |      |
| Result(+)                        |     |      | S   | S(=) |
| Service ID                       |     |      | M   | M(=) |
| Invoke ID                        |     |      | M   | M(=) |
| Local device address             |     |      | M   | M(=) |
| Local device flags               |     |      | M   | M(=) |
| Local device state               |     |      | M   | M(=) |
| Local device unique ID           |     |      | M   | M(=) |
| Device UID for R-port1           |     |      | M   | M(=) |
| Device UID for R-port2           |     |      | M   | M(=) |
| Local device MAC address         |     |      | M   | M(=) |
| Local device R-port1 information |     |      | M   | M(=) |
| Local device R-port2 information |     |      | M   | M(=) |
| Local device protocol version    |     |      | M   | M(=) |
| Local device type                |     |      | M   | M(=) |
| Local device description         |     |      | M   | M(=) |
| FamilyRes frame waiting time     |     |      | M   | M(=) |

| Parameter name             | Req | Ind | Rsp | Cnf  |
|----------------------------|-----|-----|-----|------|
| AdvThis frame waiting time |     |     | M   | M(=) |
| AckRNMS frame waiting time |     |     | M   | M(=) |
| Ring state change timeout  |     |     | M   | M(=) |
| Result(-)                  |     |     | S   | S(=) |
| Service ID                 |     |     | M   | M(=) |
| Invoke ID                  |     |     | M   | M(=) |
| Status Code                |     |     | M   | M(=) |

**Argument**

The argument conveys the service specific parameters of the service request.

**Service ID**

This parameter is defined in 7.1.

**Invoke ID**

This parameter is defined in 7.1.

**Result(+)**

This parameter indicates that the service request succeeded.

**Service ID**

This parameter is defined in 7.1.

**Invoke ID**

This parameter is defined in 7.1.

**Local device address**

This parameter is defined in 7.1.

**Local device flags**

This parameter contains the device flags of the RRP device.

**Local device state**

This parameter contains the device state of the RRP device.

**Local device unique ID**

This parameter contains the device unique ID of the RRP device.

**Device UID for R-port1**

This parameter contains the UID of the device that is linked through the R-port1.

**Device UID for R-port2**

This parameter contains the UID of the device that is linked through the R-port2.

**Local device MAC address**

This parameter is defined in 7.1.

**Local device R-port1 information**

This parameter contains the R-port1 information of the RRP device.

**Local device R-port2 information**

This parameter contains the R-port2 information of the RRP device.

**Local device protocol version**

This parameter contains the protocol version of the RRP device.

**Local device type**

This parameter is defined in 7.1.

**Local device description**

This parameter is defined in 7.1.

**FamilyRes frame waiting time**

This parameter is defined in 7.1.

**AdvThis frame waiting time**

This parameter is defined in 7.1.

**AckRNMS frame waiting time**

This parameter is defined in 7.1.

**Ring state change timeout**

This parameter is defined in 7.1.

**Result(-)**

This parameter indicates that the service request failed.

**Service ID**

This parameter is defined in 7.1.

**Invoke ID**

This parameter is defined in 7.1.

**Status Code**

This parameter is defined in 7.1.

**7.3 Get network information**

This service is used to obtain network information from the RRP device.

The parameters of this service are specified in Table 10.

**Table 10 – Parameters of get network information service**

| Parameter name             | Req | Ind  | Rsp | Cnf  |
|----------------------------|-----|------|-----|------|
| Argument                   | M   | M(=) |     |      |
| Service ID                 | M   | M(=) |     |      |
| Invoke ID                  | M   | M(=) |     |      |
| Result(+)                  |     |      | S   | S(=) |
| Service ID                 |     |      | M   | M(=) |
| Invoke ID                  |     |      | M   | M(=) |
| Network topology           |     |      | M   | M(=) |
| Collision count            |     |      | M   | M(=) |
| Device count               |     |      | M   | M(=) |
| Topology change count      |     |      | M   | M(=) |
| Last topology change time  |     |      | M   | M(=) |
| RNMP device UID            |     |      | M   | M(=) |
| RNMS device UID            |     |      | M   | M(=) |
| LNM device UID for R-port1 |     |      | M   | M(=) |
| LNM device UID for R-port2 |     |      | M   | M(=) |
| Network flags              |     |      | M   | M(=) |
| Result(-)                  |     |      | S   | S(=) |

| Parameter name | Req | Ind | Rsp | Cnf  |
|----------------|-----|-----|-----|------|
| Service ID     |     |     | M   | M(=) |
| Invoke ID      |     |     | M   | M(=) |
| Status Code    |     |     | M   | M(=) |

**Argument**

The argument conveys the service specific parameters of the service request.

**Service ID**

This parameter is defined in 7.1.

**Invoke ID**

This parameter is defined in 7.1.

**Result(+)**

This parameter indicates that the service request succeeded.

**Service ID**

This parameter is defined in 7.1.

**Invoke ID**

This parameter is defined in 7.1.

**Network topology**

This parameter contains the type of network topology.

**Collision count**

This parameter contains the device address collision count for remote devices.

**Device count**

This parameter contains the total number of devices on the network.

**Topology change count**

This parameter contains the topology change count.

**Last topology change time**

This parameter contains the date and time when the network topology was last changed.

**RNMP device UID**

This parameter contains the UID of the device selected as the RNMP on the network.

**RNMS device UID**

This parameter contains the UID of the device selected as the RNMS on the network.

**LNM device UID for R-port1**

This parameter contains the UID of the device selected as the LNM in the R-port1 direction.

**LNM device UID for R-port2**

This parameter contains the UID of the device selected as the LNM in the R-port2 direction.

**Network flags**

This parameter contains the flags for events that occurred in the network.

**Result(-)**

This parameter indicates that the service request failed.

**Service ID**

This parameter is defined in 7.1.

#### Invoke ID

This parameter is defined in 7.1.

#### Status Code

This parameter is defined in 7.1.

### 7.4 Get path table information

This service is used to obtain path table information from the RRP device. The path table is managed by the RRP protocol machine in the form of an array table filled with the path information of each RRP device on the network. The maximum size of the path table is a function of MAX\_ADDR as follows:

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

NOTE The MAX\_ADDR holds the maximum device address and is set by the RRP protocol machine. The range of this variable is 1-255. The default value of this variable is 255. This variable also indicates the maximum number of path table entries in the NMIB. The values in the range 256-65 535 are reserved.

The parameters of this service are specified in Table 11.

**Table 11 – Parameters of get path table information service**

| Parameter name                        | Req | Ind  | Rsp | Cnf  |
|---------------------------------------|-----|------|-----|------|
| Argument                              | M   | M(=) |     |      |
| Service ID                            | M   | M(=) |     |      |
| Invoke ID                             | M   | M(=) |     |      |
| Result(+)                             |     |      | S   | S(=) |
| Service ID                            |     |      | M   | M(=) |
| Invoke ID                             |     |      | M   | M(=) |
| Peer device address                   |     |      | M   | M(=) |
| Peer device hop count for R-port1     |     |      | M   | M(=) |
| Peer device hop count for R-port2     |     |      | M   | M(=) |
| Preferred R-port for peer device      |     |      | M   | M(=) |
| Destination R-port for peer device    |     |      | M   | M(=) |
| Peer device state                     |     |      | M   | M(=) |
| Peer device MAC address               |     |      | M   | M(=) |
| Peer device R-port1 information       |     |      | M   | M(=) |
| Peer device R-port2 information       |     |      | M   | M(=) |
| Peer device protocol version          |     |      | M   | M(=) |
| Peer device type                      |     |      | M   | M(=) |
| Peer device description               |     |      | M   | M(=) |
| Peer device UID                       |     |      | M   | M(=) |
| Device UID for R-port1 of peer device |     |      | M   | M(=) |
| Device UID for R-port2 of peer device |     |      | M   | M(=) |
| In net count of peer device           |     |      | M   | M(=) |
| In net time of peer device            |     |      | M   | M(=) |
| Out net count of peer device          |     |      | M   | M(=) |
| Out net time of peer device           |     |      | M   | M(=) |
| Result(-)                             |     |      | S   | S(=) |

| Parameter name | Req | Ind | Rsp | Cnf  |
|----------------|-----|-----|-----|------|
| Service ID     |     |     | M   | M(=) |
| Invoke ID      |     |     | M   | M(=) |
| Status Code    |     |     | M   | M(=) |

**Argument**

The argument conveys the service specific parameters of the service request.

**Service ID**

This parameter is defined in 7.1.

**Invoke ID**

This parameter is defined in 7.1.

**Result(+)**

This parameter indicates that the service request succeeded.

**Service ID**

This parameter is defined in 7.1.

**Invoke ID**

This parameter is defined in 7.1.

**Peer device address**

This parameter contains the value for the RRP peer device address.

**Peer device hop count for R-port1**

This parameter contains the frame forwarding counts for sending a frame from the local device to the peer device through the R-port1.

**Peer device hop count for R-port2**

This parameter contains the frame forwarding counts for sending a frame from the local device to the peer device through the R-port2.

**Preferred R-port for peer device**

This parameter contains the preferred R-port for sending a frame from the local device to the peer device without regard for the RNMP or RNMS.

**Destination R-port for peer device**

This parameter contains the destination R-port for sending a frame from the local device to the peer device.

**Peer device state**

This parameter contains the peer device state stored in the path table.

**Peer device MAC address**

This parameter contains the peer device 6 octets ISO/IEC 8802-3:2000 Ethernet MAC address stored in the path table.

**Peer device R-port1 information**

This parameter contains the peer device R-port1 information stored in the path table.

**Peer device R-port2 information**

This parameter contains the peer device R-port2 information stored in the path table.

**Peer device protocol version**

This parameter contains the peer device RRP protocol version stored in the path table.

**Peer device type**

This parameter contains the peer device application device type stored in the path table.

**Peer device description**

This parameter contains the peer device description stored in the path table.

**Peer device UID**

This parameter contains the peer device unique ID stored in the path table.

**Device UID for R-port1 of peer device**

This parameter contains the UID of the device that is linked through the R-port1 of the peer device and stored in the path table.

**Device UID for R-port2 of peer device**

This parameter contains the UID of the device that is linked through the R-port2 of the peer device and stored in the path table.

**In net count of peer device**

This parameter contains the number of times that the peer device has joined the network.

**In net time of peer device**

This parameter contains the date and time when the peer device last joined the network.

**Out net count of peer device**

This parameter contains the number of times that the peer device has been disconnected from the network.

**Out net time of peer device**

This parameter contains the date and time when the device was last disconnected from the network.

**Result(-)**

This parameter indicates that the service request failed.

**Service ID**

This parameter is defined in 7.1.

**Invoke ID**

This parameter is defined in 7.1.

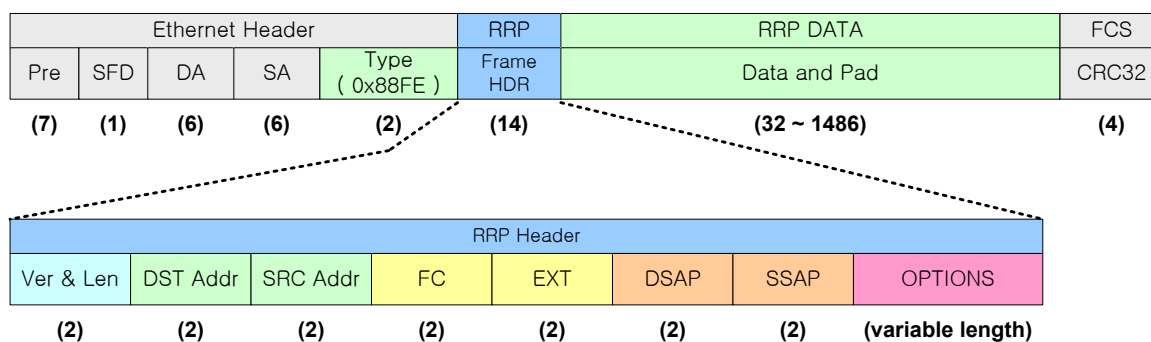
**Status Code**

This parameter is defined in 7.1.

## 8 RRP protocol specification

### 8.1 General

The encoding and decoding of the fields in the data link protocol data unit (DLPDU) is encapsulated in the data field of a MAC frame, as specified by ISO/IEC 8802-3:2000, Clause 3. The value of the Length/Type field is 0x88FE, which is authorized and registered as the protocol identification number by the IEEE Registration Authority to identify a RRP frame. Figure 18 shows the RRP DLPDU structure.



IEC 2678/11

**Figure 18 – Common MAC frame format for RRP DLPDU**

## 8.2 Ethernet header

### 8.2.1 Preamble

This field shall be coded according to ISO/IEC 8802-3:2000, Clause 3.

### 8.2.2 Start frame delimiter

This field shall be coded according to ISO/IEC 8802-3:2000, Clause 3.

### 8.2.3 Destination MAC address

This field shall be coded according to ISO/IEC 8802-3:2000, Clause 3. It specifies the device(s) for which the frame is intended, and may be an individual or multicast (including broadcast) address.

### 8.2.4 Source MAC address

This field shall be coded according to ISO/IEC 8802-3:2000, Clause 3.

### 8.2.5 Length/Type

This field shall be coded according to ISO/IEC 8802-3:2000, Clause 3 “Media access control frame structure.” To be identified as a RRP frame, the value of the Length/Type field is set to 0x88FE, which is authorized and registered as the protocol identification number for RRP by the IEEE Registration Authority. Every frame with a value other than 0x88FE is identical to the frame in ISO/IEC 8802-3:2000, Clause 3, and is processed as a RRP fieldbus sporadic data frame. For RRP, the value shall be set according to Table 12.

**Table 12 – RRP Length/Type field**

| Value (hexadecimal) | Meaning |
|---------------------|---------|
| 0x88FE              | RRP-PDU |

## 8.3 Encoding of RRP\_FrameHDR

### 8.3.1 Version and length

This field indicates the protocol version and the length for the RRP protocol. It shall be coded as data type Unsigned16, and the individual bits shall have the following meaning:

#### Bit 0 – 10: length

The value indicates the number of octets of the frame including the FCS field.

### Bit 11 – 15: version

The version is represented by two bits for the major version and three bits for the minor version. This field shall be coded with the values according to Table 13.

**Table 13 – Version**

| Field Name | Position    | Meaning                    |
|------------|-------------|----------------------------|
| Major      | Bit 14 – 15 | RRP protocol major version |
| Minor      | Bit 11 – 13 | RRP protocol minor version |

## 8.3.2 DST\_addr

### 8.3.2.1 General

This field indicates the destination RRP device identifier of the node to which the frame is sent. It shall be coded as data type Unsigned16 and set according to Table 14.

**Table 14 – DST\_addr**

| Value (hexadecimal) | Meaning                              |
|---------------------|--------------------------------------|
| 0xFFFF              | broadcast address                    |
| 0xFFFE              | network control address (C_NCM_ADDR) |
| 0xFFFD – 0xFFDE     | user-defined multicast address       |
| 0xFFDD              | invalid address                      |
| 0x0100 – 0xFFDC     | reserved                             |
| 0x0000 – 0x00FF     | regular RRP device address           |

### 8.3.2.2 Broadcast address

If the destination RRP device identifier is 0xFFFF, the destination MAC address field contains the ISO/IEC 8802-3:2000 broadcast MAC address.

### 8.3.2.3 Network control address

The RRP protocol defines a special MAC address, 00-E0-91-02-05-99 (NCM\_MAC\_ADDR), for sharing network management information. Every message received through the NCM\_MAC\_ADDR updates the network management information.

If the destination RRP device address is 0xFFFE (NCM\_ADDR), the destination MAC address field contains NCM\_MAC\_ADDR. However, all NCM messages except NCM\_ACK\_RNMS and NCM\_CHECK\_RNMS are transmitted using NCM\_ADDR as the destination RRP device address.

### 8.3.2.4 User-defined multicast address

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

### 8.3.3 SRC\_addr

This field indicates the source RRP device address of the node that generates the frame. It shall be coded as data type Unsigned16 and set according to Table 15.

**Table 15 – SRC\_addr**

| Value<br>(hexadecimal) | Meaning                   |
|------------------------|---------------------------|
| 0x0000 –<br>0xFFFF     | source RRP device address |

### 8.3.4 Frame Control (FC)

#### 8.3.4.1 General

This field indicates the frame control information. It shall be coded as data type Unsigned16. It is a bit set encoded as follows:

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

#### Bit 0 — 7: Network Control Message Type (NCMT)

This field shall be coded with the values according to Table 16.

**Table 16 – Network control message type**

| Value<br>(hexadecimal) | Meaning          | Usage     |
|------------------------|------------------|-----------|
| 0x00                   | Reserved         | —         |
| 0x01                   | NCM_FAMILY_REQ   | Mandatory |
| 0x02                   | NCM_FAMILY_RES   | Mandatory |
| 0x03                   | NCM_MEDIA_LINKED | Mandatory |
| 0x04                   | NCM_ADV_THIS     | Mandatory |
| 0x05                   | NCM_LINE_START   | Mandatory |
| 0x06                   | NCM_RING_START   | Mandatory |
| 0x07                   | NCM_ACK_RNMS     | Mandatory |
| 0x08                   | NCM_CHECK_RNMS   | Mandatory |
| 0x09-0xFF              | Reserved         | —         |

#### Bit 8 — 11: Type of Service (ToS)

This field shall be coded with the values according to Table 17.

**Table 17 – Type of service**

| Value<br>(hexadecimal) | Meaning                       | Usage     |
|------------------------|-------------------------------|-----------|
| 0x00                   | Network Control Message (NCM) | Mandatory |
| 0x01                   | unconfirmed service request   | Optional  |
| 0x02-0x0F              | Reserved                      | —         |

#### Bit 12 — 13: Priority (PRI)

This field shall be coded with the values according to Table 18.

**Table 18 – Priority**

| Value<br>(hexadecimal) | Meaning          |
|------------------------|------------------|
| 0x00                   | lowest priority  |
| 0x01-0x02              | ...              |
| 0x03                   | highest priority |

**Bit 14: Reserved (RES)**

This field shall be set to 0.

**Bit 15: Validation of Extension code (VoE)**

This field shall be coded with the values according to Table 19.

**Table 19 – Validation of extension code**

| Value<br>(decimal) | Meaning             |
|--------------------|---------------------|
| 0                  | EXT Code is invalid |
| 1                  | EXT Code is valid   |

**8.3.4.2 Network control message type**

This field indicates the type of network control message. The individual values shall have the following meanings:

**0x01: NCM\_FAMILY\_REQ**

This value is used to ask the device newly connected through an R-port if it is a RRP device. This message is transmitted through the R-port that generated the PHY link-up events. This message shall be not forwarded to the other port.

**0x02: NCM\_FAMILY\_RES**

This value is used to confirm whether the recipient is a RRP device when the recipient receives the NCM\_FAMILY\_REQ message from the newly linked device. This message is transmitted through the R-port used to receive the NCM\_FAMILY\_REQ message. This message shall be not forwarded to the other port.

**0x03: NCM\_MEDIA\_LINKED**

This value is used to indicate that a new RRP link has been established through the R-port. This message is transmitted through the newly activated R-port. When the recipient receives this message, the recipient increments the hop count in the frame and forwards the frame through the other R-port. This message is discarded by the LNM or the device that generated the message.

**0x04: NCM\_ADV\_THIS**

This value is used to transmit the recipient's local device information when the recipient receives the NCM\_MEDIA\_LINKED message from the newly linked device. This message is transmitted through the R-port used to receive the NCM\_MEDIA\_LINKED message.

**0x05: NCM\_LINE\_START**

This value is used to advise that the network topology has been automatically configured as a line network. This message is initiated by the RRP protocol machine whose state changed to LNM.

**0x06: NCM\_RING\_START**

This value is used to advise that the network topology has been automatically configured as a ring network. This message is initiated and transmitted through both R-ports by the RRP protocol machine the state of which changed to RNMP.

**0x07: NCM\_ACK\_RNMS**

This value is used by the RNMS device to advise that the RNMS has been successfully selected. The NCM\_ACK\_RNMS message is transmitted from the RNMS to the RNMP through the received R-port of NCM\_RING\_START or NCM\_CHECK\_RNMS message.

**0x08: NCM\_CHECK\_RNMS**

This value is used to request the NCM\_ACK\_RNMS message from the RNMS device in the case where the RNMP device wants to check the network integrity. The NCM\_CHECK\_RNMS message is transmitted from the RNMP to the RNMS with the address of RNMS through both of R-port1 and R-port2.

NOTE All the message of RRP network with a specific destination address is using one R-port for the shortest path for efficiency. But the NCM\_CHECK\_RNMS message is also used to check integrity of RRP network, NCM\_CHECK\_RNMS message is transmitted through both of R-port1 and R-port2.

**8.3.4.3 Type of Service (ToS)**

This field indicates the type of data link (DL) service. A value of 0x00 indicates a network control message among RRP devices, and 0x01 indicates an unconfirmed service request among users.

**8.3.4.4 Priority (PRI)**

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

**8.3.4.5 Validation of extension code (VoE)**

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

**8.4 Encoding of data and pad****8.4.1 General**

This field indicates the data field received from a user. Network control messages are used to transfer network control messages among RRP devices. Eight message types are provided to share the network information.

**8.4.2 Encoding of FamilyReq**

The FamilyReq frame is encoded as specified in Table 20.

**Table 20 – Encoding of FamilyReq frame**

| No. | Parameter name         | Data type        | Octet offset | Octet length | Description                                   |
|-----|------------------------|------------------|--------------|--------------|-----------------------------------------------|
| 1   | Device address         | Unsigned16       | 0            | 2            | Local device address                          |
| 2   | Device flags           | Unsigned16       | 2            | 2            | Local device flags                            |
| 3   | Device type            | Unsigned16       | 4            | 2            | Local device type                             |
| 4   | Hop count              | Unsigned16       | 6            | 2            | Hop count                                     |
| 5   | Device UID             | UniqueDeviceID64 | 8            | 8            | Local device unique ID                        |
| 6   | Device UID for R-port1 | UniqueDeviceID64 | 16           | 8            | Unique ID of device connected through R-port1 |
| 7   | Device UID for R-port2 | UniqueDeviceID64 | 24           | 8            | Unique ID of device connected through R-port2 |
| 8   | MAC address            | Unsigned48       | 32           | 6            | Local device MAC address                      |
| 9   | Reserved0              | Unsigned16       | 38           | 2            | Reserved, – set to 0.                         |
| 10  | R-port1 information    | Unsigned8        | 40           | 1            | Local device R-port1 information              |

| No. | Parameter name      | Data type     | Octet offset | Octet length | Description                      |
|-----|---------------------|---------------|--------------|--------------|----------------------------------|
| 11  | R-port2 information | Unsigned8     | 41           | 1            | Local device R-port2 information |
| 12  | Device state        | Unsigned8     | 42           | 1            | Local device state               |
| 13  | Protocol version    | Unsigned8     | 43           | 1            | Local device protocol version    |
| 14  | Device description  | VisibleString | 44           | 16           | Device description string        |
| 15  | Reserved1           | Unsigned32    | 60           | 4            | Reserved, – set to 0.            |

### 8.4.3 Encoding of FamilyRes

The FamilyRes frame is encoded as specified in Table 21.

**Table 21 – Encoding of FamilyRes frame**

| No. | Parameter name         | Data type        | Octet offset | Octet length | Description                                   |
|-----|------------------------|------------------|--------------|--------------|-----------------------------------------------|
| 1   | Device address         | Unsigned16       | 0            | 2            | Local device address                          |
| 2   | Device flags           | Unsigned16       | 2            | 2            | Local device flags                            |
| 3   | Device type            | Unsigned16       | 4            | 2            | Local device type                             |
| 4   | Hop count              | Unsigned16       | 6            | 2            | Hop count                                     |
| 5   | Device UID             | UniqueDeviceID64 | 8            | 8            | Local device unique ID                        |
| 6   | Device UID for R-port1 | UniqueDeviceID64 | 16           | 8            | Unique ID of device connected through R-port1 |
| 7   | Device UID for R-port2 | UniqueDeviceID64 | 24           | 8            | Unique ID of device connected through R-port2 |
| 8   | MAC address            | Unsigned48       | 32           | 6            | Local device MAC address                      |
| 9   | Reserved0              | Unsigned16       | 38           | 2            | Reserved, – set to 0.                         |
| 10  | R-port1 information    | Unsigned8        | 40           | 1            | Local device R-port1 information              |
| 11  | R-port2 information    | Unsigned8        | 41           | 1            | Local device R-port2 information              |
| 12  | Device state           | Unsigned8        | 42           | 1            | Local device state                            |
| 13  | Protocol version       | Unsigned8        | 43           | 1            | Local device protocol version                 |
| 14  | Device description     | VisibleString    | 44           | 16           | Device description string                     |
| 15  | Reserved1              | Unsigned32       | 60           | 4            | Reserved, – set to 0.                         |

### 8.4.4 Encoding of MediaLinked

The MediaLinked frame is encoded as specified in Table 22.

**Table 22 – Encoding of MediaLinked frame**

| No. | Parameter name         | Data type        | Octet offset | Octet length | Description                                   |
|-----|------------------------|------------------|--------------|--------------|-----------------------------------------------|
| 1   | Device address         | Unsigned16       | 0            | 2            | Local device address                          |
| 2   | Device flags           | Unsigned16       | 2            | 2            | Local device flags                            |
| 3   | Device type            | Unsigned16       | 4            | 2            | Local device type                             |
| 4   | Hop count              | Unsigned16       | 6            | 2            | Hop count                                     |
| 5   | Device UID             | UniqueDeviceID64 | 8            | 8            | Local device unique ID                        |
| 6   | Device UID for R-port1 | UniqueDeviceID64 | 16           | 8            | Unique ID of device connected through R-port1 |

| No. | Parameter name         | Data type        | Octet offset | Octet length | Description                                   |
|-----|------------------------|------------------|--------------|--------------|-----------------------------------------------|
| 7   | Device UID for R-port2 | UniqueDeviceID64 | 24           | 8            | Unique ID of device connected through R-port2 |
| 8   | MAC address            | Unsigned48       | 32           | 6            | Local device MAC address                      |
| 9   | Reserved0              | Unsigned16       | 38           | 2            | Reserved, – set to 0.                         |
| 10  | R-port1 information    | Unsigned8        | 40           | 1            | Local device R-port1 information              |
| 11  | R-port2 information    | Unsigned8        | 41           | 1            | Local device R-port2 information              |
| 12  | Device state           | Unsigned8        | 42           | 1            | Local device state                            |
| 13  | Protocol version       | Unsigned8        | 43           | 1            | Local device protocol version                 |
| 14  | Device description     | VisibleString    | 44           | 16           | Device description string                     |
| 15  | Reserved1              | Unsigned32       | 60           | 4            | Reserved, – set to 0.                         |

#### 8.4.5 Encoding of AdvThis

The AdvThis frame is encoded as specified in Table 23.

**Table 23 – Encoding of AdvThis frame**

| No. | Parameter name         | Data type        | Octet offset | Octet length | Description                                   |
|-----|------------------------|------------------|--------------|--------------|-----------------------------------------------|
| 1   | Device address         | Unsigned16       | 0            | 2            | Local device address                          |
| 2   | Device flags           | Unsigned16       | 2            | 2            | Local device flags                            |
| 3   | Device type            | Unsigned16       | 4            | 2            | Local device type                             |
| 4   | Hop count              | Unsigned16       | 6            | 2            | Hop count                                     |
| 5   | Device UID             | UniqueDeviceID64 | 8            | 8            | Local device unique ID                        |
| 6   | Device UID for R-port1 | UniqueDeviceID64 | 16           | 8            | Unique ID of device connected through R-port1 |
| 7   | Device UID for R-port2 | UniqueDeviceID64 | 24           | 8            | Unique ID of device connected through R-port2 |
| 8   | MAC address            | Unsigned48       | 32           | 6            | Local device MAC address                      |
| 9   | Reserved0              | Unsigned16       | 38           | 2            | Reserved, – set to 0.                         |
| 10  | R-port1 information    | Unsigned8        | 40           | 1            | Local device R-port1 information              |
| 11  | R-port2 information    | Unsigned8        | 41           | 1            | Local device R-port2 information              |
| 12  | Device state           | Unsigned8        | 42           | 1            | Local device state                            |
| 13  | Protocol version       | Unsigned8        | 43           | 1            | Local device protocol version                 |
| 14  | Device description     | VisibleString    | 44           | 16           | Device description string                     |
| 15  | Reserved1              | Unsigned32       | 60           | 4            | Reserved, – set to 0.                         |

#### 8.4.6 Encoding of LineStart

The LineStart frame is encoded as specified in Table 24

**Table 24 – Encoding of LineStart frame**

| No. | Parameter name | Data type  | Octet offset | Octet length | Description          |
|-----|----------------|------------|--------------|--------------|----------------------|
| 1   | Device address | Unsigned16 | 0            | 2            | Local device address |
| 2   | Device flags   | Unsigned16 | 2            | 2            | Local device flags   |

| No. | Parameter name             | Data type        | Octet offset | Octet length | Description                                              |
|-----|----------------------------|------------------|--------------|--------------|----------------------------------------------------------|
| 3   | Device type                | Unsigned16       | 4            | 2            | Local device type                                        |
| 4   | Hop count                  | Unsigned16       | 6            | 2            | Hop count                                                |
| 5   | Device UID                 | UniqueDeviceID64 | 8            | 8            | Local device unique ID                                   |
| 6   | Device UID for R-port1     | UniqueDeviceID64 | 16           | 8            | Unique ID of device connected through R-port1            |
| 7   | Device UID for R-port2     | UniqueDeviceID64 | 24           | 8            | Unique ID of device connected through R-port2            |
| 8   | MAC address                | Unsigned48       | 32           | 6            | Local device MAC address                                 |
| 9   | Reserved0                  | Unsigned16       | 38           | 2            | Reserved, – set to 0.                                    |
| 10  | R-port1 information        | Unsigned8        | 40           | 1            | Local device R-port1 information                         |
| 11  | R-port2 information        | Unsigned8        | 41           | 1            | Local device R-port2 information                         |
| 12  | Device state               | Unsigned8        | 42           | 1            | Local device state                                       |
| 13  | Protocol version           | Unsigned8        | 43           | 1            | Local device protocol version                            |
| 14  | Device description         | VisibleString    | 44           | 16           | Device description string                                |
| 15  | Reserved1                  | Unsigned32       | 60           | 4            | Reserved, – set to 0.                                    |
| 16  | Topology                   | Unsigned8        | 64           | 1            | RRP network topology                                     |
| 17  | Collision count            | Unsigned8        | 65           | 1            | Device address collision count between remote devices    |
| 18  | Device count               | Unsigned16       | 66           | 2            | Device count for the network segment                     |
| 19  | Topology change count      | Unsigned16       | 68           | 2            | Network topology change count                            |
| 20  | Network flags              | Unsigned16       | 70           | 2            | Network event flags                                      |
| 21  | Last topology change time  | TIMEOFDAY        | 72           | 6            | Date and time when the network topology last was changed |
| 22  | Reserved2                  | Unsigned16       | 78           | 2            | Reserved, – set to 0.                                    |
| 23  | RNMP device UID            | UniqueDeviceID64 | 80           | 8            | UID of the RNMP device                                   |
| 24  | RNMS device UID            | UniqueDeviceID64 | 88           | 8            | UID of the RNMS device                                   |
| 25  | LNM device UID for R-port1 | UniqueDeviceID64 | 96           | 8            | UID of the LNM device in R-port1 direction               |
| 26  | LNM device UID for R-port2 | UniqueDeviceID64 | 104          | 8            | UID of the LNM device in R-port2 direction               |

#### 8.4.7 Encoding of RingStart

The RingStart frame is encoded as specified in Table 25.

**Table 25 – Encoding of RingStart frame**

| No. | Parameter name         | Data type        | Octet offset | Octet length | Description                                   |
|-----|------------------------|------------------|--------------|--------------|-----------------------------------------------|
| 1   | Device address         | Unsigned16       | 0            | 2            | Local device address                          |
| 2   | Device flags           | Unsigned16       | 2            | 2            | Local device flags                            |
| 3   | Device type            | Unsigned16       | 4            | 2            | Local device type                             |
| 4   | Hop count              | Unsigned16       | 6            | 2            | Hop count                                     |
| 5   | Device UID             | UniqueDeviceID64 | 8            | 8            | Local device unique ID                        |
| 6   | Device UID for R-port1 | UniqueDeviceID64 | 16           | 8            | Unique ID of device connected through R-port1 |

| No. | Parameter name             | Data type        | Octet offset | Octet length | Description                                              |
|-----|----------------------------|------------------|--------------|--------------|----------------------------------------------------------|
| 7   | Device UID for R-port2     | UniqueDeviceID64 | 24           | 8            | Unique ID of device connected through R-port2            |
| 8   | MAC address                | Unsigned48       | 32           | 6            | Local device MAC address                                 |
| 9   | Reserved0                  | Unsigned16       | 38           | 2            | Reserved, – set to 0.                                    |
| 10  | R-port1 information        | Unsigned8        | 40           | 1            | Local device R-port1 information                         |
| 11  | R-port2 information        | Unsigned8        | 41           | 1            | Local device R-port2 information                         |
| 12  | Device state               | Unsigned8        | 42           | 1            | Local device state                                       |
| 13  | Protocol version           | Unsigned8        | 43           | 1            | Local device protocol version                            |
| 14  | Device description         | VisibleString    | 44           | 16           | Device description string                                |
| 15  | Reserved1                  | Unsigned32       | 60           | 4            | Reserved, – set to 0.                                    |
| 16  | Topology                   | Unsigned8        | 64           | 1            | RRP network topology                                     |
| 17  | Collision count            | Unsigned8        | 65           | 1            | Device address collision count between remote devices    |
| 18  | Device count               | Unsigned16       | 66           | 2            | Device count for the network segment                     |
| 19  | Topology change count      | Unsigned16       | 68           | 2            | Network topology change count                            |
| 20  | Network flags              | Unsigned16       | 70           | 2            | Network event flags                                      |
| 21  | Last topology change time  | TIMEOFDAY        | 72           | 6            | Date and time when the network topology last was changed |
| 22  | Reserved2                  | Unsigned16       | 78           | 2            | Reserved, – set to 0.                                    |
| 23  | RNMP device UID            | UniqueDeviceID64 | 80           | 8            | UID of the RNMP device                                   |
| 24  | RNMS device UID            | UniqueDeviceID64 | 88           | 8            | UID of the RNMS device                                   |
| 25  | LNM device UID for R-port1 | UniqueDeviceID64 | 96           | 8            | UID of the LNM device in the R-port1 direction           |
| 26  | LNM device UID for R-port2 | UniqueDeviceID64 | 104          | 8            | UID of the LNM device in the R-port2 direction           |

#### 8.4.8 Encoding of AckRNMS

The AckRNMS frame is encoded as specified in Table 26.

**Table 26 – Encoding of AckRNMS frame**

| No. | Parameter name         | Data type        | Octet offset | Octet length | Description                                   |
|-----|------------------------|------------------|--------------|--------------|-----------------------------------------------|
| 1   | Device address         | Unsigned16       | 0            | 2            | Local device address                          |
| 2   | Device flags           | Unsigned16       | 2            | 2            | Local device flags                            |
| 3   | Device type            | Unsigned16       | 4            | 2            | Local device type                             |
| 4   | Hop count              | Unsigned16       | 6            | 2            | Hop count                                     |
| 5   | Device UID             | UniqueDeviceID64 | 8            | 8            | Local device unique ID                        |
| 6   | Device UID for R-port1 | UniqueDeviceID64 | 16           | 8            | Unique ID of device connected through R-port1 |
| 7   | Device UID for R-port2 | UniqueDeviceID64 | 24           | 8            | Unique ID of device connected through R-port2 |
| 8   | MAC address            | Unsigned48       | 32           | 6            | Local device MAC address                      |
| 9   | Reserved0              | Unsigned16       | 38           | 2            | Reserved, – set to 0.                         |
| 10  | R-port1 information    | Unsigned8        | 40           | 1            | Local device R-port1 information              |
| 11  | R-port2 information    | Unsigned8        | 41           | 1            | Local device R-port2 information              |

| No. | Parameter name     | Data type     | Octet offset | Octet length | Description                   |
|-----|--------------------|---------------|--------------|--------------|-------------------------------|
| 12  | Device state       | Unsigned8     | 42           | 1            | Local device state            |
| 13  | Protocol version   | Unsigned8     | 43           | 1            | Local device protocol version |
| 14  | Device description | VisibleString | 44           | 16           | Device description string     |
| 15  | Reserved1          | Unsigned32    | 60           | 4            | Reserved, – set to 0.         |

#### 8.4.9 Encoding of CheckRNMS

The CheckRNMS frame is encoded as specified in Table 27.

**Table 27 – Encoding of CheckRNMS frame**

| No. | Parameter name         | Data type        | Octet offset | Octet length | Description                                   |
|-----|------------------------|------------------|--------------|--------------|-----------------------------------------------|
| 1   | Device address         | Unsigned16       | 0            | 2            | Local device address                          |
| 2   | Device flags           | Unsigned16       | 2            | 2            | Local device flags                            |
| 3   | Device type            | Unsigned16       | 4            | 2            | Local device type                             |
| 4   | Hop count              | Unsigned16       | 6            | 2            | Hop count                                     |
| 5   | Device UID             | UniqueDeviceID64 | 8            | 8            | Local device unique ID                        |
| 6   | Device UID for R-port1 | UniqueDeviceID64 | 16           | 8            | Unique ID of device connected through R-port1 |
| 7   | Device UID for R-port2 | UniqueDeviceID64 | 24           | 8            | Unique ID of device connected through R-port2 |
| 8   | MAC address            | Unsigned48       | 32           | 6            | Local device MAC address                      |
| 9   | Reserved0              | Unsigned16       | 38           | 2            | Reserved, – set to 0.                         |
| 10  | R-port1 information    | Unsigned8        | 40           | 1            | Local device R-port1 information              |
| 11  | R-port2 information    | Unsigned8        | 41           | 1            | Local device R-port2 information              |
| 12  | Device state           | Unsigned8        | 42           | 1            | Local device state                            |
| 13  | Protocol version       | Unsigned8        | 43           | 1            | Local device protocol version                 |
| 14  | Device description     | VisibleString    | 44           | 16           | Device description string                     |
| 15  | Reserved1              | Unsigned32       | 60           | 4            | Reserved, – set to 0.                         |

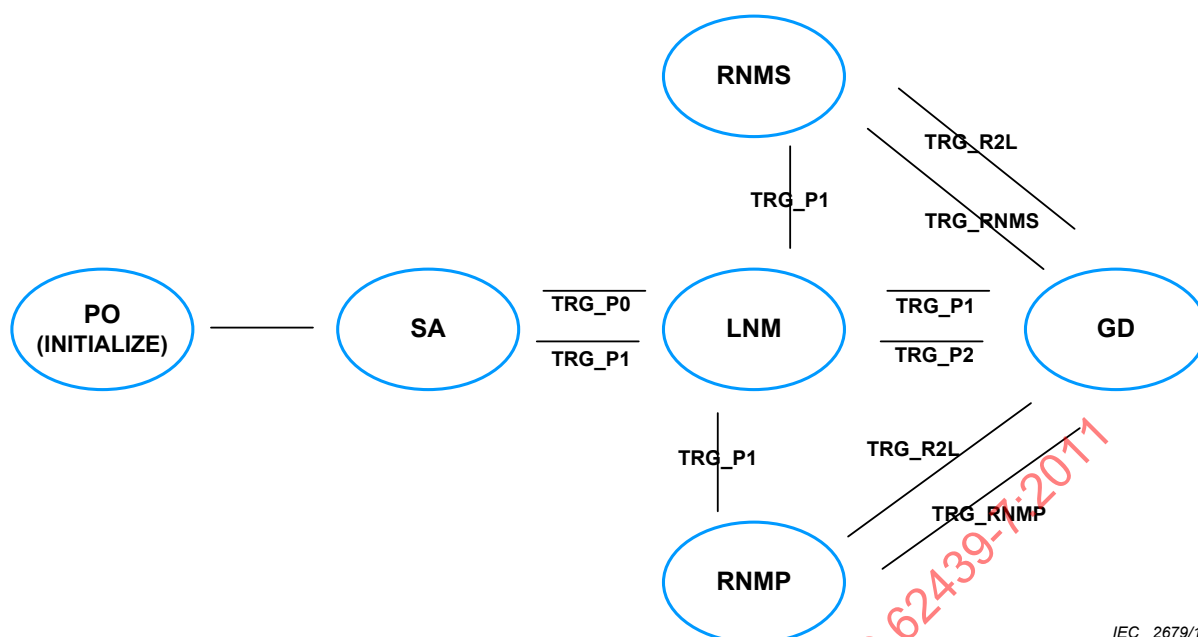
#### 8.5 Frame Check Sequence (FCS)

The FCS construction, polynomial, and expected residual are identical to those in ISO/IEC 8802-3:2000, Clause 3.

### 9 RRP protocol machine

#### 9.1 Protocol state machine description

The RRP protocol machine is shown in Figure 19.



IEC 2679/11

**Figure 19 – RRP protocol state machine**

The text below is an explanation of the overall actions performed in the states. If there is a difference in interpretation between this text and the state machine, then the state machine governs. A RRP device has six states: PO, SA, LNM, GD, RNMP, and RNMS.

#### **PO (power on, initialize)**

This state means that the local device is starting up its power supply and running a self initialization procedure. After initial process is completed, the device changes its state to SA.

#### **SA (standalone)**

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

#### **LNM (line network manager)**

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

#### **GD (general device)**

This state means that the local device is connected to the network through both its R-ports. Both frame forward functions are enabled in the GD device. However, the frame forward functions in the GD device are suspended until the device receives a LineStart message (NCM\_LINE\_START) or a RingStart message (NCM\_RING\_START).

#### **RNMP (primary ring network manager)**

This state means that the local device has been automatically selected as the primary ring network manager in a ring network. The RNMP device selects one of its neighbouring devices as the secondary ring network manager (RNMS) using the RingStart message (NCM\_RING\_START). The RNMP device disables the frame forward function in the RNMS direction but keeps the frame forward function in the other direction enabled.

#### **RNMS (secondary ring network manager)**

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

Each state has transactions. These transactions are defined as follows:

#### **TRG\_P0**

This transaction indicates that both ports of the device are LINK\_DOWN or not acknowledged LINK\_UP.

#### **TRG\_P1**

This transaction indicates that only one port of the device is LINK\_UP and LINK\_UP has been acknowledged.

#### **TRG\_P2**

This transaction indicates that both ports of the device are LINK\_UP and LINK\_UP has been acknowledged.

#### **TRG\_R2L**

This transaction indicates that the device has received NCM\_LINE\_START message.

#### **TRG\_RNMP**

This transaction indicates that the GD has elected as a RNMP.

#### **TRG\_RNMS**

This transaction indicates that the GD has selected as a RNMS by the RNMP.

### **9.2 Local parameters and variables for protocol state**

#### **9.2.1 General**

Local parameters and variables include local device information, network information, and path table information. Local data link information, such as the device address and the status of each R-port, is stored in the local device information. The network topology and the network-related variables are stored in the network information, and the device profile and path information of the other devices on the network are summarized in the path table.

#### **9.2.2 Variables to support local device information management**

To maintain the network topology, each device manages a device database that includes the local device information and information about other devices. Table 28 shows a list of device information management variables.

**Table 28 – Variables to support device information management**

| <b>Name</b>  | <b>Data type</b>  | <b>Description</b>                            |
|--------------|-------------------|-----------------------------------------------|
| DEV_ADDR     | Unsigned16        | Local device address                          |
| DEV_FLAG     | Unsigned16        | Local device flags                            |
| DEV_STATE    | Unsigned8         | Local device state                            |
| DEV_UID      | UniqueDeviceID64  | Local device unique ID                        |
| DEV_UID_RP1  | UniqueDeviceID64  | Unique ID of device connected through R-port1 |
| DEV_UID_RP2  | UniqueDeviceID64  | Unique ID of device connected through R-port2 |
| MAC_ADDR     | Unsigned48        | Local device MAC address                      |
| PORT1_INFO   | Unsigned8         | Local device R-port1 information              |
| PORT2_INFO   | Unsigned8         | Local device R-port2 information              |
| PROTOCOL_VER | Unsigned8         | Local device protocol version                 |
| DEV_TYPE     | Unsigned16        | Local device type                             |
| DEV_DESC     | VisibleString[16] | Device description string                     |
| HOP_CNT      | Unsigned16        | Hop count                                     |

### 9.2.3 Variables to support network information management

The network information is managed automatically by the RRP protocol machine. The network information variables are summarized in Table 29.

**Table 29 – Variables to support managing network information**

| Name        | Data type        | Description                                |
|-------------|------------------|--------------------------------------------|
| RRP_NET_TPG | Unsigned8        | RRP network topology                       |
| UID_RNMP    | UniqueDeviceID64 | UID of the RNMP device                     |
| UID_RNMS    | UniqueDeviceID64 | UID of the RNMS device                     |
| UID_LNM_RP1 | UniqueDeviceID64 | UID of the LNM device in R-port1 direction |
| UID_LNM_RP2 | UniqueDeviceID64 | UID of the LNM device in R-port2 direction |

### 9.2.4 Variables to support device path information management

The path table is managed by the RRP protocol machine. The variables for a path table item are defined in Table 30.

**Table 30 – Variables to support device path information management**

| Name              | Data type         | Description                                                           |
|-------------------|-------------------|-----------------------------------------------------------------------|
| path-DEV_ADDR     | Unsigned16        | Peer device address                                                   |
| path-HOP_CNT_RP1  | Unsigned16        | Hop count in R-port1 direction                                        |
| path-HOP_CNT_RP2  | Unsigned16        | Hop count in R-port2 direction                                        |
| path-PREFER_RP    | Unsigned8         | R-port with the smaller hop count value to the peer device            |
| path-DST_RP       | Unsigned8         | Selected R-port for sending a frame to the destination device address |
| path-DEV_STATE    | Unsigned8         | Peer device state                                                     |
| path-MAC_ADDR     | Unsigned48        | Peer device ISO/IEC 8802-3:2000 MAC address                           |
| path-PORT1_INFO   | Unsigned8         | Peer device local R-port1 information                                 |
| path-PORT2_INFO   | Unsigned8         | Peer device local R-port2 information                                 |
| path-PROTOCOL_VER | Unsigned8         | Peer device protocol version                                          |
| path-DEV_TYPE     | Unsigned16        | Peer device application device type                                   |
| path-DEV_DESC     | VisibleString[16] | Peer device description                                               |
| path-DEV_UID      | UniqueDeviceID64  | Peer device UID                                                       |
| path-DEV_UID_RP1  | UniqueDeviceID64  | Device UID of the device connected through the peer device R-port1    |
| path-DEV_UID_RP2  | UniqueDeviceID64  | Device UID of the device connected through the peer device R-port2    |

### 9.2.5 Variables of Received RRP Frame

The variables of the Received RRP frame are listed in Table 31.

**Table 31 – Variables of Received RRP Frame**

| Name          | Data type  | Description                      |
|---------------|------------|----------------------------------|
| rcv-DEV_ADDR  | Unsigned16 | Device address in received frame |
| rcv-DEV_STATE | Unsigned8  | Device state in received frame   |

| Name           | Data type        | Description                              |
|----------------|------------------|------------------------------------------|
| rcv-DEV_UID    | UniqueDeviceID64 | Unique ID of device in received frame    |
| rcv-PORT1_INFO | Unsigned8        | R-port1 information in received frame    |
| rcv-PORT2_INFO | Unsigned8        | R-port2 information in received frame    |
| rcv-HOP_CNT    | Unsigned16       | Hop count in received frame              |
| rcv-UID_RNMP   | UniqueDeviceID64 | UID of the RNMP device in received frame |
| rcv-UID_RNMS   | UniqueDeviceID64 | UID of the RNMS device in received frame |

### 9.2.6 Local variables for protocol state

The local variables for the protocol state are listed in Table 32.

**Table 32 – Local variables for protocol state**

| Name                 | Data type  | Description                                                |
|----------------------|------------|------------------------------------------------------------|
| Link_status          | Boolean    | PHY link status of local device                            |
| RRP_FamilyReqT       | Unsigned32 | FamilyRes frame waiting time<br>Default value is 3 ms      |
| RRP_MediaLinkedT     | Unsigned32 | AdvThis frame waiting time<br>Default value is 3 ms        |
| RRP_AckRN MST        | Unsigned32 | AckRNMS frame waiting time<br>Default value is 3 ms        |
| RRP_ChkRN MST        | Unsigned32 | ChkRNMS frame waiting time<br>Default value is 3 ms        |
| RRP_ChangeRingStateT | Unsigned32 | Ring state change timeout<br>Default value is 3 ms         |
| RRP_RetryCount       | Unsigned32 | RRP network integrity check count<br>Default value is 3 ms |
| RRP_ChkIntegrityT    | Unsigned32 | ChkRNMS time interval time<br>Default value is 3 ms        |

### 9.2.7 Constants for protocol state

The constants for the protocol state are listed in Table 33.

**Table 33 – Constants for protocol state**

| Constant        | Description                                   | Value |
|-----------------|-----------------------------------------------|-------|
| PHY_LINK_UP     | Event generated when PHY link is up           | TRUE  |
| PHY_LINK_DOWN   | Event generated when PHY link is down         | FALSE |
| TRUE            | Return value 1                                | 1     |
| FALSE           | Return value 0                                | 0     |
| INVALID_UID     | Invalid unique identification                 | 0     |
| INVALID_R_PORT  | Invalid R-port state                          | 0     |
| PORT_LINK_DOWN  | R-port link down                              | 0x01  |
| PORT_CFM_FAMILY | Received Family_Frame.Cnf from R-port         | 0x02  |
| PORT_WAIT_ADV   | Waiting for AdvThis_Frame.Cnf from R-port     | 0x04  |
| PORT_WAIT_ML    | Waiting for MediaLinked_Frame.Ind from R-port | 0x08  |
| PORT_CFM        | Device state confirm                          | 0x10  |

### 9.3 State transitions

The state transitions of the RRP protocol state machine are specified in Table 34.

**Table 34 – RRP State transitions**

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                            | Next state |
|-----|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 1   | PO            | POWER-ON or RESET<br>/<br>=><br>INIT_DEV_INFO( )<br>INIT_NET_INFO( )<br>INIT_PATH_INFO( )<br>MAC-RESET.req{ }<br>Ph-RESET.req{ }<br>Set_Block_Port(INVALID_R_PORT)<br>Clear_Port_Info(R-port1)<br>Clear_Port_Info(R-port2)<br>DEV_STATE := SA<br>STORE_PATH_INFO(DEV_ADDR) | SA         |
| 2   | SA            | Phy_Link_Change.Ind(R-port, Link_status)<br>/R-port == R-port1<br>&& Link_status == PHY_LINK_UP<br>=><br>Set_Block_Port(R-port2)<br>Start_Timer(RRP_FamilyReqT)<br>Family_Frame.Req(R-port)                                                                                | SA         |
| 3   | SA            | Phy_Link_Change.Ind(R-port, Link_status)<br>/R-port == R-port2<br>&& Link_status == PHY_LINK_UP<br>=><br>Set_Block_Port(R-port1)<br>Start_Timer(RRP_FamilyReqT)<br>Family_Frame.Req(R-port)                                                                                | SA         |
| 4   | SA            | Family_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/<br>=><br>Set_Neighbor_UID(R-port, rcv-DEV_UID)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Family_Frame.Res(R-port)                                                                                                       | SA         |
| 5   | SA            | Timer(RRP_FamilyReqT) expired<br>/<br>=><br>Start_Timer(RRP_FamilyReqT)<br>Family_Frame.Req(R-port)                                                                                                                                                                        | SA         |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                                                                                                                                             | Next state |
|-----|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 6   | SA            | Family_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/<br>=><br>Stop_Timer(RRP_FamilyReqT)<br>Set_Neighbor_UID(R-port, rcv-DEV_UID)<br>Set_Port_Info(R-port, PORT_CFM_FAMILY)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Start_Timer(RRP_MediaLinkedT)<br>MediaLinked_Frame.Reg(R-port)                                                                                                                                                                                          | SA         |
| 7   | SA            | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID == Get_Neighbor_UID(R-port)<br>&& Chk_Port_Info(R-port, PORT_CFM_FAMILY) == FALSE<br>=><br>Ignore                                                                                                                                                                                                                                                                                                  | SA         |
| 8   | SA            | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID == Get_Neighbor_UID(R-port)<br>&& Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ML) == TRUE<br>=><br>Set_Port_Info(R-port, PORT_CFM)<br>Set_Block_Port(INVALID_R_PORT)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>RRP_NET_TPG := NET_TPG_LINE<br>Set_LNM_UID(R-port, DEV_UID)<br>DEV_STATE := LNM<br>AdvThis_Frame.Res(R-port)<br>RRP_Line_Start_Frame.Reg(R-port) | LNМ        |
| 9   | SA            | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID == Get_Neighbor_UID(R-port)<br>&& Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ML) == FALSE<br>=><br>Set_Port_Info(R-port, PORT_WAIT_ADV)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>AdvThis_Frame.Res(R-port)                                                                                                                                                    | SA         |
| 10  | SA            | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID != Get_Neighbor_UID(R-port)<br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>AdvThis_Frame.Res(R-port)                                                                                                                                                                                                                                                                                                    | SA         |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Next state |
|-----|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 11  | SA            | Timer(RRP_MediaLinkedT) expired<br><br>/<br><br>=><br>Start_Timer(RRP_MediaLinkedT)<br>MediaLinked_Frame.Reg(R-port)                                                                                                                                                                                                                                                                                                                                                                                                                                     | SA         |
| 12  | SA            | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == FALSE<br><br>=><br>Ignore                                                                                                                                                                                                                                                                                                                                                                                      | SA         |
| 13  | SA            | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ADV) == TRUE<br>&& (rcv-PORT1_INFO == PORT_LINK_DOWN<br>   rcv-PORT2_INFO == PORT_LINK_DOWN)<br><br>=><br>Stop_Timer(RRP_MediaLinkedT)<br>Set_Port_Info(R-port, PORT_CFM)<br>Set_Block_Port(INVALID_R_PORT)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Set_LNM_UID(R-port, DEV_UID)<br>DEV_STATE := LNM<br>RRP_NET_TPG := NET_TPG_LINE<br>RRP_Line_Start_Frame.Reg(R-port) | LNLM       |
| 14  | SA            | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ADV) == FALSE<br>&& (rcv-PORT1_INFO == PORT_LINK_DOWN<br>   rcv-PORT2_INFO == PORT_LINK_DOWN)<br><br>=><br>Stop_Timer(RRP_MediaLinkedT)<br>Set_Port_Info(R-port, PORT_WAIT_ML)<br>STORE_PATH_INFO(rcv-DEV_ADDR)                                                                                                                                                     | SA         |
| 15  | SA            | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& rcv-PORT1_INFO != PORT_LINK_DOWN<br>&& rcv-PORT2_INFO != PORT_LINK_DOWN<br><br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)                                                                                                                                                                                                                                                                                  | SA         |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                     | Next state |
|-----|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 16  | SA            | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>=><br>Set_Block_Port(INVALID_R_PORT)<br>Clear_Port_Info(R-port1)<br>Clear_Port_Info(R-port2)<br>INIT_PATH_INFO( )<br>STORE_PATH_INFO(DEV_ADDR)                                                         | SA         |
| 17  | LNM           | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_UP<br>=><br>Start_Timer(RRP_FamilyReqT)<br>Family_Frame.Req(R-port)                                                                                                                                            | LNM        |
| 18  | LNM           | Family_Frame.Ind (R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/<br>=><br>Set_Neighbor_UID(R-port, rcv-DEV_UID)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Family_Frame.Res(R-port)                                                                                                               | LNM        |
| 19  | LNM           | Timer(RRP_FamilyReqT) expired<br>/<br>=><br>Start_Timer(RRP_FamilyReqT)<br>Family_Frame.Req(R-port)                                                                                                                                                                                 | LNM        |
| 20  | LNM           | Family_Frame.Cnf (R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/<br>=><br>Stop_Timer(RRP_FamilyReqT)<br>Set_Neighbor_UID(R-port, rcv-DEV_UID)<br>Set_Port_Info(R-port, PORT_CFM_FAMILY)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Start_Timer(RRP_MediaLinkedT)<br>MediaLinked_Frame.Req(R-port) | LNM        |
| 21  | LNM           | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID == Get_Neighbor_UID(R-port)<br>&& Chk_Port_Info(R-port, PORT_CFM_FAMILY) == FALSE<br>=><br>Ignore                                                                                                          | LNM        |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                                                                                | Next state |
|-----|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 22  | LNM           | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID == Get_Neighbor_UID(R-port)<br>&& Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ML) == TRUE<br>=><br>Set_Port_Info(R-port, PORT_CFM)<br>DEV_STATE := GD<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Clear_LNM_UID(R-port)<br>Forward_Frame(R-port, MediaLinked_Frame.Ind)<br>AdvThis_Frame.Res(R-port) | GD         |
| 23  | LNM           | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID == Get_Neighbor_UID(R-port)<br>&& Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ML) == FALSE<br>=><br>Set_Port_Info(R-port, PORT_WAIT_ADV)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, MediaLinked_Frame.Ind)<br>AdvThis_Frame.Res(R-port)                                       | LNM        |
| 24  | LNM           | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID != Get_Neighbor_UID(R-port)<br>&& rcv-DEV_UID != DEV_UID<br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, MediaLinked_Frame.Ind)<br>AdvThis_Frame.Res(R-port)                                                                                                                                                          | LNM        |
| 25  | LNM           | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID == DEV_UID<br>=><br>Ignore                                                                                                                                                                                                                                                                                                            | LNM        |
| 26  | LNM           | Timer(RRP_MediaLinkedT) expired<br>/<br>=><br>Start_Timer(RRP_MediaLinkedT)<br>MediaLinked_Frame.Reg(R-port)                                                                                                                                                                                                                                                                                                   | LNM        |
| 27  | LNM           | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == FALSE<br>=><br>Ignore                                                                                                                                                                                                                                                    | LNM        |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Next state |
|-----|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 28  | LNM           | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ADV) == TRUE<br>&& (rcv-DEV_UID == UID_LNM_RP1<br>   rcv-DEV_UID == UID_LNM_RP2)<br>&& rcv-DEV_UID == Get_Neighbor_UID(R-port)<br>=><br>Stop_Timer(RRP_MediaLinkedT)<br>Set_Port_Info(R-port, PORT_CFM)<br>DEV_STATE := GD<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Clear_LNM_UID(R-port)<br>Forward_Frame(R-port, AdvThis_Frame.Cnf)                                   | GD         |
| 29  | LNM           | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ADV) == TRUE<br>&& (rcv-DEV_UID != UID_LNM_RP1<br>&& rcv-DEV_UID != UID_LNM_RP2)<br>&& (rcv-PORT1_INFO == PORT_LINK_DOWN<br>   rcv-PORT2_INFO == PORT_LINK_DOWN)<br>=><br>Stop_Timer(RRP_MediaLinkedT)<br>Set_Port_Info(R-port, PORT_CFM)<br>DEV_STATE := GD<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Clear_LNM_UID(R-port)<br>Forward_Frame(R-port, AdvThis_Frame.Cnf) | GD         |
| 30  | LNM           | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ADV) == FALSE<br>&& (rcv-DEV_UID == UID_LNM_RP1<br>   rcv-DEV_UID == UID_LNM_RP2)<br>&& rcv-DEV_UID == Get_Neighbor_UID(R-port)<br>=><br>Stop_Timer(RRP_MediaLinkedT)<br>Set_Port_Info(R-port, PORT_WAIT_ML)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, AdvThis_Frame.Cnf)                                                                          | LNM        |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Next state |
|-----|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 31  | LNM           | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& Chk_Port_Info(R-port, PORT_WAIT_ADV) == FALSE<br>&& (rcv-DEV_UID != UID_LNM_RP1<br>&& rcv-DEV_UID != UID_LNM_RP2)<br>&& (rcv-PORT1_INFO == PORT_LINK_DOWN<br>   rcv-PORT2_INFO == PORT_LINK_DOWN)<br>=><br>Stop_Timer(RRP_MediaLinkedT)<br>Set_Port_Info(R-port, PORT_WAIT_ML)<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, AdvThis_Frame.Cnf) | LNM        |
| 32  | LNM           | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& (rcv-DEV_UID == UID_LNM_RP1<br>   rcv-DEV_UID == UID_LNM_RP2)<br>&& rcv-DEV_UID != Get_Neighbor_UID(R-port)<br>&& rcv-DEV_UID != DEV_UID<br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, AdvThis_Frame.Cnf)                                                                                                                                 | LNM        |
| 33  | LNM           | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& (rcv-DEV_UID != UID_LNM_RP1<br>&& rcv-DEV_UID != UID_LNM_RP2)<br>&& (rcv-PORT1_INFO != PORT_LINK_DOWN<br>&& rcv-PORT2_INFO != PORT_LINK_DOWN)<br>&& rcv-DEV_UID != DEV_UID<br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, AdvThis_Frame.Cnf)                                                                                               | LNM        |
| 34  | LNM           | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br><br>/Chk_Port_Info(R-port, PORT_CFM_FAMILY) == TRUE<br>&& rcv-DEV_UID == DEV_UID<br>=><br>RRP_NET_TPG := NET_TPG_RING<br>Start_Timer(RRP_ChangeRingStateT)                                                                                                                                                                                                                                                            | LNM        |
| 35  | LNM           | Timer(RRP_ChangeRingStateT) expired<br>/<br>=><br>Start_Timer(RRP_ChangeRingStateT)                                                                                                                                                                                                                                                                                                                                                                                                                              | LNM        |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                                             | Next state |
|-----|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 36  | LNM           | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& Chk_Port_Info(R-port, PORT_CFM) == FALSE<br>&& R-port == R-port1<br>=><br>Clear_Port_Info(R-port)<br>DELETE_PATH_INFO(R-port)<br>RRP_Line_Start_Frame.Reg(R-port2)                                                                                                                          | LNM        |
| 37  | LNM           | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& Chk_Port_Info(R-port, PORT_CFM) == FALSE<br>&& R-port == R-port2<br>=><br>Clear_Port_Info(R-port)<br>DELETE_PATH_INFO(R-port)<br>RRP_Line_Start_Frame.Reg(R-port1)                                                                                                                          | LNM        |
| 38  | LNM           | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& Chk_Port_Info(R-port, PORT_CFM) == TRUE<br>=><br>INIT_DEV_INFO( )<br>INIT_NET_INFO( )<br>INIT_PATH_INFO( )<br>Set_Block_Port(INVALID_R_PORT)<br>Clear_Port_Info(R-port1)<br>Clear_Port_Info(R-port2)<br>DEV_STATE := SA<br>STORE_PATH_INFO(DEV_ADDR)<br>MAC-RESET.req{ }<br>Ph-RESET.req{ } | SA         |
| 39  | LNM           | RRP_Line_Start_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/<br>=><br>Store_LNM_UID(R-port, rcv-DEV_UID)<br>STORE_PATH_INFO(rcv-DEV_ADDR)                                                                                                                                                                                                                               | LNM        |
| 40  | GD            | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID == DEV_UID<br>=><br>Ignore                                                                                                                                                                                                                                                                         | GD         |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                   | Next state |
|-----|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 41  | GD            | MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/rcv-DEV_UID != DEV_UID<br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, MediaLinked_Frame.Ind)<br>AdvThis_Frame.Res(R-port)                                                                                                                           | GD         |
| 42  | GD            | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br>/rcv-DEV_UID == DEV_UID<br>=><br>RRP_NET_TPG := NET_TPG_RING<br>Start_Timer(RRP_ChangeRingStateT)                                                                                                                                      | GD         |
| 43  | GD            | AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR,<br>rcv-PORT1_INFO, rcv-PORT2_INFO)<br>/rcv-DEV_UID != DEV_UID<br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, AdvThis_Frame.Cnf)                                                                                                                             | GD         |
| 44  | GD            | Timer(RRP_ChangeRingStateT) expired<br>/RRP_NET_TPG == NET_TPG_RING<br>&& CheckRNMP(UID_RNMP, UID_RNMS) == TRUE<br>=><br>Stop_Timer(RRP_ChangeRingStateT)<br>DEV_STATE := RNMP<br>Clear_LNM_UID(0)<br>RRP_Ring_Start_Frame.Req(R-port2, UID_RNMP, UID_RNMS)<br>Start_Timer(RRP_AckRNMS)                                           | RNMP       |
| 45  | GD            | Timer(RRP_ChangeRingStateT) expired<br>/RRP_NET_TPG == NET_TPG_RING<br>   CheckRNMP(UID_RNMP, UID_RNMS) == FALSE<br>=><br>Stop_Timer(RRP_ChangeRingStateT)                                                                                                                                                                        | GD         |
| 46  | GD            | RRP_Ring_Start_Frame.Ind(R-port, rcv-DEV_ADDR, rcv-UID_RNMP,<br>rcv-UID_RNMS)<br>/rcv-UID_RNMS == DEV_UID<br>=><br>DEV_STATE := RNMS<br>Clear_LNM_UID(0)<br>UID_RNMP := rcv-UID_RNMP<br>UID_RNMS := rcv-UID_RNMS<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, RRP_Ring_Start_Frame.Ind)<br>AckRNMS_Frame.Res(R-port) | RNMS       |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                | Next state |
|-----|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 47  | GD            | RRP_Ring_Start_Frame.Ind(R-port, rcv-DEV_ADDR, rcv-UID_RNMP,<br>rcv-UID_RNMS)<br><br>/rcv-UID_RNMS != DEV_UID<br><br>=><br>Clear_LNM_UID(0)<br>UID_RNMP := rcv-UID_RNMP<br>UID_RNMS := rcv-UID_RNMS<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, RRP_Ring_Start_Frame.Ind)                                                        | GD         |
| 48  | GD            | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& R-port == R-port1<br>=><br>Clear_Port_Info(R-port)<br>DEV_STATE := LNM<br>DELETE_PATH_INFO(R-port)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, DEV_UID)<br>RRP_Line_Start_Frame.Reg(R-port2) | LNM        |
| 49  | GD            | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& R-port == R-port2<br>=><br>Clear_Port_Info(R-port)<br>DEV_STATE := LNM<br>DELETE_PATH_INFO(R-port)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, DEV_UID)<br>RRP_Line_Start_Frame.Reg(R-port1) | LNM        |
| 50  | GD            | RRP_Line_Start_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/<br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, rcv-DEV_UID)<br>Forward_Frame(R-port, RRP_Line_Start_Frame.Ind)                                                          | GD         |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                | Next state |
|-----|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 51  | RNMP          | AckRNMS_Frame.Cnf(R-port, rcv-DEV_UID)<br>/rcv-DEV_UID == UID_RNMS<br>=><br>Stop_Timer(RRP_AckRNMST)                                                                                                                                                                                                                                           | RNMP       |
| 52  | RNMP          | AckRNMS_Frame.Cnf(R-port, rcv-DEV_UID)<br>/rcv-DEV_UID != UID_RNMS<br>=><br>ignore                                                                                                                                                                                                                                                             | RNMP       |
| 53  | RNMP          | Timer(RRP_AckRNMST) expired<br>/<br>=><br>CheckRNMS_Frame.Reg(UID_RNMS)<br>Start_Timer(RRP_AckRNMST)                                                                                                                                                                                                                                           | RNMP       |
| 54  | RNMP          | RRP_Ring_Start_Frame.Ind(R-port, rcv-DEV_ADDR, rcv-UID_RNMP,<br>rcv-UID_RNMS)<br>/rcv-UID_RNMP == DEV_UID<br>=><br>Ignore                                                                                                                                                                                                                      | RNMP       |
| 55  | RNMP          | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& R-port == R-port1<br>=><br>Clear_Port_Info(R-port)<br>DEV_STATE := LNM<br>DELETE_PATH_INFO(R-port)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, DEV_UID)<br>RRP_Line_Start_Frame.Reg(R-port2) | LNM        |
| 56  | RNMP          | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& R-port == R-port2<br>=><br>Clear_Port_Info(R-port)<br>DEV_STATE := LNM<br>DELETE_PATH_INFO(R-port)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, DEV_UID)<br>RRP_Line_Start_Frame.Reg(R-port1) | LNM        |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                | Next state |
|-----|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 57  | RNMP          | RRP_Line_Start_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/<br>=><br>DEV_STATE := GD<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, rcv-DEV_UID)<br>Forward_Frame(R-port, RRP_Line_Start_Frame.Ind)                                       | GD         |
| 58  | RNMS          | RRP_Ring_Start_Frame.Ind(R-port, rcv-DEV_ADDR, rcv-UID_RNMP, rcv-UID_RNMS)<br>/rcv-UID_RNMS == DEV_UID<br>=><br>Clear_LNM_UID(0)<br>UID_RNMP := rcv-UID_RNMP<br>UID_RNMS := rcv-UID_RNMS<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>Forward_Frame(R-port, RRP_Ring_Start_Frame.Ind)<br>AckRNMS_Frame.Res(R-port)                                      | RNMS       |
| 59  | RNMS          | CheckRNMS_Frame.Ind(R-port, rcv-DEV_ADDR)<br>/<br>=><br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>AckRNMS_Frame.Res(R-port)                                                                                                                                                                                                                             | RNMS       |
| 60  | RNMS          | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& R-port == R-port1<br>=><br>Clear_Port_Info(R-port)<br>DEV_STATE := LNM<br>DELETE_PATH_INFO(R-port)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, DEV_UID)<br>RRP_Line_Start_Frame.Req(R-port2) | LNМ        |

| No. | Current state | Event<br>/Condition<br>=>Action                                                                                                                                                                                                                                                                                                                | Next state |
|-----|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 61  | RNMS          | Phy_Link_Change.Ind(R-port, Link_status)<br>/Link_status == PHY_LINK_DOWN<br>&& R-port == R-port2<br>=><br>Clear_Port_Info(R-port)<br>DEV_STATE := LNM<br>DELETE_PATH_INFO(R-port)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, DEV_UID)<br>RRP_Line_Start_Frame.Reg(R-port1) | LNM        |
| 62  | RNMS          | RRP_Line_Start_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)<br>/<br>=><br>DEV_STATE := GD<br>STORE_PATH_INFO(rcv-DEV_ADDR)<br>UID_RNMP := INVALID_UID<br>UID_RNMS := INVALID_UID<br>RRP_NET_TPG := NET_TPG_LINE<br>Store_LNM_UID(R-port, rcv-DEV_UID)<br>Forward_Frame(R-port, RRP_Line_Start_Frame.Ind)                                       | GD         |

## 9.4 Function descriptions

The RRP functions shall be according to Table 35.

**Table 35 – RRP Function descriptions**

| Function name                                       | Operations                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Phy_Link_Change.Ind(R-port, Link_status)            | <p>Receive a local link change indication of the R-port from PHY</p> <p>R-port := port that caused the local link change indication.</p> <p>Link_status := PHY_LINK_UP or PHY_LINK_DOWN (depends on the local link change indication)</p>                                          |
| Family_Frame.Req(R-port)                            | <p>Create and send RRP Family Request frame</p> <p>Destination MAC Address := NCM_MAC_ADDR</p> <p>DST_addr := C_NCM_ADDR</p> <p>SRC_addr := DEV_ADDR</p> <p>FC.NCMT := NCM_FAMILY_REQ</p> <p>RRP-DATA := Local_Device_Information</p> <p>SendFrame(R-port, RRP_HDR, RRP-DATA)</p>  |
| Family_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR) | <p>Receive RRP Family Request frame</p> <p>R-port := port that received the RRP Family Request frame</p> <p>rcv-DEV_UID := the DEV_UID of the received frame</p> <p>rcv-DEV_ADDR := the DEV_ADDR of the received frame</p>                                                         |
| Family_Frame.Res(R-port)                            | <p>Create and send RRP Family Response frame</p> <p>Destination MAC Address := NCM_MAC_ADDR</p> <p>DST_addr := C_NCM_ADDR</p> <p>SRC_addr := DEV_ADDR</p> <p>FC.NCMT := NCM_FAMILY_RES</p> <p>RRP-DATA := Local_Device_Information</p> <p>SendFrame(R-port, RRP_HDR, RRP-DATA)</p> |
| Family_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR) | <p>Receive RRP Family Response frame</p> <p>R-port := port that received the RRP Family Response frame</p> <p>rcv-DEV_UID := the DEV_UID of the received frame</p> <p>rcv-DEV_ADDR := the DEV_ADDR of the received frame</p>                                                       |

| Function name                                                                        | Operations                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MediaLinked_Frame.Req(R-port)                                                        | <p>Create and send RRP Media Linked frame</p> <p>Destination MAC Address := NCM_MAC_ADDR<br/> DST_addr := C_NCM_ADDR<br/> SRC_addr := DEV_ADDR<br/> FC.NCMT := NCM_MEDIA_LINKED<br/> RRP-DATA := Local_Device_Information</p> <p>SendFrame(R-port, RRP_HDR, RRP-DATA)</p>                                                          |
| MediaLinked_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)                             | <p>Receive RRP Media Linked frame</p> <p>R-port := port that received the RRP Media Linked frame<br/> rcv-DEV_UID := the DEV_UID of the received frame<br/> rcv-DEV_ADDR := the DEV_ADDR of the received frame</p>                                                                                                                 |
| AdvThis_Frame.Res(R-port)                                                            | <p>Create and send RRP Adv This frame</p> <p>Destination MAC Address := NCM_MAC_ADDR<br/> DST_addr := C_NCM_ADDR<br/> SRC_addr := DEV_ADDR<br/> FC.NCMT := NCM_ADV_THIS<br/> RRP-DATA := Local_Device_Information</p> <p>SendFrame(R-port, RRP_HDR, RRP-DATA)</p>                                                                  |
| AdvThis_Frame.Cnf(R-port, rcv-DEV_UID, rcv-DEV_ADDR, rcv-PORT1_INFO, rcv-PORT2_INFO) | <p>Receive RRP Adv This frame</p> <p>R-port := port that received the RRP Adv This frame<br/> rcv-DEV_UID := the DEV_UID of the received frame<br/> rcv-DEV_ADDR := the DEV_ADDR of the received frame<br/> rcv-PORT1_INFO := the PORT1_INFO of the received frame<br/> rcv-PORT2_INFO := the PORT2_INFO of the received frame</p> |
| RRP_Line_Start_Frame.Req(R-port)                                                     | <p>Create and send RRP Line Start frame</p> <p>Destination MAC Address := NCM_MAC_ADDR<br/> DST_addr := C_NCM_ADDR<br/> SRC_addr := DEV_ADDR<br/> FC.NCMT := NCM_LINE_START<br/> RRP-DATA := Local_Device_Information , Net_Information</p> <p>SendFrame(R-port, RRP_HDR, RRP-DATA)</p>                                            |
| RRP_Line_Start_Frame.Ind(R-port, rcv-DEV_UID, rcv-DEV_ADDR)                          | <p>Receive RRP Line Start frame</p> <p>R-port := port that received the RRP Line Start frame<br/> rcv-DEV_UID := the DEV_UID of the received frame<br/> rcv-DEV_ADDR := the DEV_ADDR of the received frame</p>                                                                                                                     |

| Function name                                                              | Operations                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RRP_Ring_Start_Frame.Req(R-port, UID_RNMP, UID_RNMS)                       | <p>Create and send RRP Ring Start frame</p> <p>Destination MAC Address := NCM_MAC_ADDR<br/> DST_addr := C_NCM_ADDR<br/> SRC_addr := DEV_ADDR<br/> FC.NCMT := NCM_RING_START<br/> RRP-DATA := Local_Device_Information, Net_Information</p> <p>SendFrame(R-port, RRP_HDR, RRP-DATA)</p> |
| RRP_Ring_Start_Frame.Ind(R-port, rcv-DEV_ADDR, rcv-UID_RNMP, rcv-UID_RNMS) | <p>Receive RRP Ring Start frame</p> <p>R-port := port that received the RRP Ring Start frame<br/> rcv-DEV_ADDR := the DEV_ADDR of the received frame<br/> rcv-UID_RNMP := the UID_RNMP of the received frame<br/> rcv-UID_RNMS := the UID_RNMS of the received frame</p>               |
| AckRNMS_Frame.Res(R-port)                                                  | <p>Create and send RRP ACK RNMS frame</p> <p>DST_addr := UID_RNMP<br/> SRC_addr := DEV_ADDR<br/> FC.NCMT := NCM_ACK_RNMS<br/> RRP-DATA := Local_Device_Information</p> <p>SendFrame(R-port, RRP_HDR, RRP-DATA)</p>                                                                     |
| AckRNMS_Frame.Cnf(R-port, rcv-DEV_UID)                                     | <p>Receive RRP ACK RNMS frame</p> <p>R-port := port that received the RRP ACK RNMS frame<br/> rcv-DEV_UID := the DEV_UID of the received frame</p>                                                                                                                                     |
| CheckRNMS_Frame.Req(UID_RNMS)                                              | <p>Create and send RRP Check RNMS frame</p> <p>DST_addr := UID_RNMS<br/> SRC_addr := DEV_ADDR<br/> FC.NCMT := NCM_CHECK_RNMS<br/> RRP-DATA := Local_Device_Information</p> <p>SendFrame(R-port, RRP_HDR, RRP-DATA)</p>                                                                 |
| CheckRNMS_Frame.Ind(R-port, rcv-DEV_ADDR)                                  | <p>Receive RRP Retry RNMS frame</p> <p>R-port := port that received the RRP ACK RNMS frame<br/> rcv-DEV_ADDR := the DEV_ADDR of the received frame</p>                                                                                                                                 |

| Function name                         | Operations                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Forward_Frame(R-port, Received_Frame) | <p>Forward received frame from R-port</p> <p>R-port := port that received the frame<br/>Received_Frame := the received frame</p> <p>Increments the hop counts in the received frame by 1</p> <p>If R-port == 1, then SendReceivedFrame(R-port2, Received_Frame)</p> <p>If R-port == 2, then SendReceivedFrame(R-port1, Received_Frame)</p> |
| INIT_DEV_INFO()                       | Initialize variables to support local device information management                                                                                                                                                                                                                                                                        |
| Set_Port_Info(R-port, Status)         | <p>Function to set the port status of R-port to Status</p> <p>If R-port == 1, then PORT1_INFO := (PORT1_INFO   Status)</p> <p>If R-port == 2, then PORT2_INFO := (PORT2_INFO   Status)</p>                                                                                                                                                 |
| Clear_Port_Info(R-port)               | <p>Function to clear the port status of R-port</p> <p>If R-port == 1, then PORT1_INFO := PORT_LINK_DOWN</p> <p>If R-port == 2, then PORT2_INFO := PORT_LINK_DOWN</p>                                                                                                                                                                       |
| Chk_Port_Info(R-port, Status)         | <p>Function to check the port status of R-port</p> <p>If (Get_Port_Info(R-port) &amp; Status) == Status, then return TRUE</p> <p>else return FALSE</p>                                                                                                                                                                                     |
| Get_Port_Info(R-port)                 | <p>Function to get the port status of R-port</p> <p>If R-port == 1, then return PORT1_INFO</p> <p>If R-port == 2, then return PORT2_INFO</p>                                                                                                                                                                                               |
| Set_Neighbor_UID(R-port, rcv-DEV_UID) | <p>Function to set the UID of the device linked through the R-port</p> <p>If R-port == 1, then DEV_UID_RP1 := rcv-DEV_UID</p> <p>If R-port == 2, then DEV_UID_RP2 := rcv-DEV_UID</p>                                                                                                                                                       |
| Get_Neighbor_UID(R-port)              | <p>Function to get the UID of the device linked through the R-port</p> <p>If R-port == 1, then return DEV_UID_RP1</p> <p>If R-port == 2, then return DEV_UID_RP2</p>                                                                                                                                                                       |
| INIT_NET_INFO()                       | Initialize variables to support network information management                                                                                                                                                                                                                                                                             |
| Set_LNM_UID(R-port, dev_UID)          | <p>Function to set the UID of the device selected as the LNM in the R-port direction</p> <p>If R-port == 1, then UID_LNM_RP2 := dev_UID</p> <p>If R-port == 2, then UID_LNM_RP1 := dev_UID</p>                                                                                                                                             |
| Store_LNM_UID(R-port, dev_UID)        | <p>Function to store the UID of the device selected as the LNM in the R-port direction</p> <p>If R-port == 1, then UID_LNM_RP1 := dev_UID</p> <p>If R-port == 2, then UID_LNM_RP2 := dev_UID</p>                                                                                                                                           |
| Clear_LNM_UID(R-port)                 | <p>Function to clear the UID of the device selected as the LNM in the R-port direction</p> <p>If R-port == 0, then UID_LNM_RP1 := INVALID_UID and UID_LNM_RP2 := INVALID_UID</p> <p>If R-port == 1, then UID_LNM_RP1 := INVALID_UID</p> <p>If R-port == 2, then UID_LNM_RP2 := INVALID_UID</p>                                             |

| Function name                             | Operations                                                                                                                                                                                                                                                  |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INIT_PATH_INFO( )                         | Initialize variables to support device path information management                                                                                                                                                                                          |
| STORE_PATH_INFO(dev_ADDR)                 | Add or update path information using DEV_INFO in RRP-DATA from received frame<br><br>If SearchPathTable(dev_ADDR) == TRUE, then update path information in path table<br><br>If SearchPathTable(dev_ADDR) == FALSE, then add path information to path table |
| DELETE_PATH_INFO(R-port)                  | Function to delete path information related to R-port                                                                                                                                                                                                       |
| CheckRNMP(UID_RNMP, UID_RNMS)             | Function to check RNMP and select RNMS<br><br>If the DEV_UID is the biggest UID in the path table, then UID_RNMP := DEV_UID and UID_RNMS := DEV_UID_RP1 and return TRUE<br><br>If the DEV_UID is not the biggest UID in the path table, then return FALSE   |
| Set_Block_Port(R-port)                    | Function to set R-port to blocked state<br><br>If R-port == 0, then set R-port1 and R-port2 to active state<br>If R-port == 1, then set R-port1 to blocked state<br>If R-port == 2, then set R-port2 to blocked state                                       |
| MAC-RESET.req{ }                          | Function to reset the MAC                                                                                                                                                                                                                                   |
| Ph-RESET.req{ }                           | Function to reset the PHY                                                                                                                                                                                                                                   |
| Start_Timer(t)                            | Function to start the timer                                                                                                                                                                                                                                 |
| Stop_Timer(t)                             | Function to stop the timer                                                                                                                                                                                                                                  |
| SendFrame(R-port, RRP_HDR, RRP-DATA)      | Function to send the RRP frame<br>R-port := port for sending a frame                                                                                                                                                                                        |
| SendReceivedFrame(R-port, Received_Frame) | Function to forward the RRP frame to the other R-port.<br>R-port := port that received the frame<br>If R-port == 1, then send the RRP frame to R-port2<br>If R-port == 2, then send the RRP frame to R-port1                                                |
| SearchPathTable(dev_ADDR)                 | Function to search the path information for dev_ADDR<br><br>If path information for dev_ADDR exists in the path table, then return TRUE<br><br>If path information for dev_ADDR does not exist in the path table, then return FALSE                         |

## 10 RRP Management Information Base (MIB)

```
-- *****
IEC-62439-7-MIB DEFINITIONS ::= BEGIN

-- *****
-- Imports
-- *****

IMPORTS
    OBJECT-IDENTITY,
    OBJECT-TYPE,
    TimeTicks,
    Counter32,
    Unsigned32,
    Counter64,
    VISIBLE-STRING,
    INTEGER                      FROM SNMPv2-SMI
    Boolean                      FROM HOST-RESOURCES-MIB
    MacAddress                   FROM BRIDGE-MIB
    iso                          FROM RFC1155-SMI;

-- *****
-- Declaration of TIMEOFDAY
-- *****
TIMEOFDAY ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION "
        The IEC 61158-5-21 defines the structure of
        the TIMEOFDAY as a data type numeric
        identifier 12.
    "
    SYNTAX          VISIBLE STRING (SIZE (6))

-- *****
-- Root OID
-- *****
iec62439 MODULE-IDENTITY
    LAST-UPDATED "201306110000Z" -- June 11, 2013
    ORGANIZATION "IEC/SC 65C"
    CONTACT-INFO "
        International Electrotechnical Commission
        IEC Central Office
        3, rue de Varembe
        P.O. Box 131
        CH - 1211 GENEVA 20
        Switzerland
        Phone: +41 22 919 02 11
        Fax: +41 22 919 03 00
        email: info@iec.ch
    "
    DESCRIPTION "
        This MIB module defines the Network Management interfaces
        for the Redundancy Protocols defined by the IEC
        standard 62439.
    "

    REVISION "201306110000Z" -- June 11, 2013
    DESCRIPTION "
        Consistency adjustment with other parts of IEC 62439
    "

    REVISION "200811100000Z" -- November 10, 2008
    DESCRIPTION "
        Seperation of IEC 62439 into a suite of documents.
        This MIB applies to IEC 62439-7, added RRP functionality.
    "

    REVISION "200708240000Z" -- August 24, 2007
    DESCRIPTION "
        Initial version of the Network Management interface for the
        Ring-based Redundancy Protocol
    "

::= { IEC 62439 }
```

```
-- *****
-- Redundancy Protocols
-- *****

mrp          OBJECT IDENTIFIER ::= { iec62439 1 }
prp          OBJECT IDENTIFIER ::= { iec62439 2 }
crp          OBJECT IDENTIFIER ::= { iec62439 3 }
brp          OBJECT IDENTIFIER ::= { iec62439 4 }
drp          OBJECT IDENTIFIER ::= { iec62439 5 }
rrp          OBJECT IDENTIFIER ::= { iec62439 6 }

-- *****
-- Objects of the RRP Network Management
-- *****

ServiceID    OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      mandatory
    DESCRIPTION  "
                specifies information sufficient for local
                identification of the RRP device that will
                convey the service
                "
    ::= { rrp 1 }

InvokeID     OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      mandatory
    DESCRIPTION  "
                specifies the invocation of the service
                "
    ::= { rrp 2 }

DeviceAddress OBJECT-TYPE
    SYNTAX      INTEGER
    MAX-ACCESS  read-write
    STATUS      mandatory
    DESCRIPTION  "
                specifies the value for the RRP device address
                "
    ::= { rrp 3 }

DeviceFlags  OBJECT-TYPE
    SYNTAX      BITS {
                DeviceAddressCollision(0),
                DeviceStateChanged(1)
                }
    MAX-ACCESS  read-only
    STATUS      mandatory
    DESCRIPTION  "
                specifies the flags for events that occurred
                in a local device
                "
    ::= { rrp 4 }

DeviceState  OBJECT-TYPE
    SYNTAX      INTEGER {
                Invalid(0),
                SA(1),
                LNM(2),
                GD(3),
                RNMP(4),
                RNMS(5)
                }
    MAX-ACCESS  read-only
    STATUS      mandatory
    DESCRIPTION  "
                specifies the device state of the RRP device
                "
    ::= { rrp 5 }

DeviceUID    OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      mandatory
    DESCRIPTION  "
```

```

        specifies the unique 8-octet identification that
        identifies a RRP device in a network
        "
 ::= { rrp 6 }

DeviceUIDRport1 OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       mandatory
    DESCRIPTION  "
        specifies the UID of the device that is linked
        through the R-port1
        "
 ::= { rrp 7 }

DeviceUIDRport2 OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       mandatory
    DESCRIPTION  "
        specifies the UID of the device that is linked
        through the R-port2
        "
 ::= { rrp 8 }

MACAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS   read-write
    STATUS       mandatory
    DESCRIPTION  "
        specifies the MAC address of the RRP device
        "
 ::= { rrp 9 }

Rport1Information OBJECT-TYPE
    SYNTAX      BITS {
        PortLinkDown(0),
        PortCFMFamily(1),
        PortWaitADV(2),
        PortWaitML(3)
        PortCFM(4)
    }
    MAX-ACCESS   read-only
    STATUS       mandatory
    DESCRIPTION  "
        specifies the port information for R-port1
        "
 ::= { rrp 10 }

Rport2Information OBJECT-TYPE
    SYNTAX      BITS {
        PortLinkDown(0),
        PortCFMFamily(1),
        PortWaitADV(2),
        PortWaitML(3)
        PortCFM(4)
    }
    MAX-ACCESS   read-only
    STATUS       mandatory
    DESCRIPTION  "
        specifies the port information for R-port2
        "
 ::= { rrp 11 }

Version OBJECT-TYPE
    SYNTAX      INTEGER
    MAX-ACCESS   read-write
    STATUS       mandatory
    DESCRIPTION  "
        specifies the protocol version of the RRP device
        "
 ::= { rrp 12 }

DeviceType OBJECT-TYPE
    SYNTAX      INTEGER
    MAX-ACCESS   read-write
    STATUS       mandatory
    DESCRIPTION  "

```

```
-- *****
-- Objects of the RRP Network Information
-- *****
```

```

STATUS          mandatory
DESCRIPTION      "
                  specifies the type of network topology
                  "

 ::= { rrp 20 }

CollisionCnt     OBJECT-TYPE
SYNTAX           INTEGER
MAX-ACCESS       read-only
STATUS           mandatory
DESCRIPTION      "
                  specifies the device address collision count
                  for remote devices
                  "

 ::= { rrp 21 }

DeviceCnt        OBJECT-TYPE
SYNTAX           INTEGER
MAX-ACCESS       read-only
STATUS           mandatory
DESCRIPTION      "
                  specifies the total number of devices on the network
                  "

 ::= { rrp 22 }

TopologyChangeCnt OBJECT-TYPE
SYNTAX           INTEGER
MAX-ACCESS       read-only
STATUS           mandatory
DESCRIPTION      "
                  specifies the topology change count
                  "

 ::= { rrp 23 }

LastTopologyChangeTime OBJECT-TYPE
SYNTAX           TIMEOFDAY
MAX-ACCESS       read-only
STATUS           mandatory
DESCRIPTION      "
                  specifies the date and time at which the network topology
                  was last changed
                  "

 ::= { rrp 24 }

RNMPDeviceUID    OBJECT-TYPE
SYNTAX           Counter64
MAX-ACCESS       read-only
STATUS           mandatory
DESCRIPTION      "
                  specifies the UID of the device selected as the RNMP
                  on the network
                  "

 ::= { rrp 25 }

RNMSDeviceUID    OBJECT-TYPE
SYNTAX           Counter64
MAX-ACCESS       read-only
STATUS           mandatory
DESCRIPTION      "
                  specifies the UID of the device selected as the RNMS
                  on the network
                  "

 ::= { rrp 26 }

LNMDDeviceUIDRport1 OBJECT-TYPE
SYNTAX           Counter64
MAX-ACCESS       read-only
STATUS           mandatory
DESCRIPTION      "
                  specifies the UID of the device selected as the LNM
                  in the R-port1 direction
                  "

 ::= { rrp 27 }

LNMDDeviceUIDRport2 OBJECT-TYPE
SYNTAX           Counter64
MAX-ACCESS       read-only
STATUS           mandatory

```

```

DESCRIPTION      "
                  specifies the UID of the device selected as the LNM
                  in the R-port2 direction
                  "
::= { rrp 28 }

NetworkFlags      OBJECT-TYPE
SYNTAX            BITS {
                  NetworkTopologyChanged(0),
                  DeviceAddressCollisionInNetwork(1),
                  DeviceJoinedNetwork(2),
                  DeviceLeftNetwork(3)
                  }
MAX-ACCESS        read-only
STATUS            mandatory
DESCRIPTION       "
                  specifies the flags for events that occurred
                  in the network
                  "
::= { rrp 29 }

-- *****
-- Objects of the RRP Path Table Information
-- *****

PathTableInfo      OBJECT-TYPE
SYNTAX            SEQUENCE OF PathTableInfoEntry
MAX-ACCESS        read-only
STATUS            mandatory
DESCRIPTION       "
                  Path Table in the form of an array table containing
                  the path information foreach RRP device on the network
                  "
::= { rrp 30 }

PathTableEntry      OBJECT-TYPE
SYNTAX            PathTableInfoEntry
MAX-ACCESS        read-only
STATUS            mandatory
DESCRIPTION       "Row of Path Table Information"
INDEX            { PathTableIndex }
::= { PathTableInfo 1 }

PathTableInfoEntry ::= SEQUENCE {
    pathDevAddress      INTEGER,
    pathHopCntRp1       INTEGER,
    pathHopCntRp2       INTEGER,
    pathPreferRp        INTEGER,
    pathHostRp          INTEGER,
    pathDevState        INTEGER,
    pathMACAddress      MacAddress,
    pathRp1Info         INTEGER,
    pathRp2Info         INTEGER,
    pathVersion         INTEGER,
    pathDevType         INTEGER,
    pathDevDesc         VISIBLE STRING (SIZE(1..16)),
    pathDevUID          Counter64,
    pathDevUIDRp1       Counter64,
    pathDevUIDRp2       Counter64
}

END
-- *****
-- EOF
-- *****

```

## Bibliography

IEC 61158 (all parts), *Industrial communication networks – Fieldbus specifications*

IEC 61158-4-21: 2010, *Industrial communication networks – Fieldbus specifications – Part 4-21: Data-link layer protocol specification – Type 21 elements*

ISO/IEC 9314-3:1990, *Information processing systems – Fibre distributed Data Interface (FDDI) – Part 3: Physical Layer Medium Dependent (PMD)*

IEEE 802.1D:2004, *IEEE Standard for Local and Metropolitan Area Networks – Media access control (MAC) Bridges*

ANSI X3.230:1994, *Information Technology – Fibre Channel – Physical and Signaling Interface (FC-PH)*

ANSI X3.263:1995, *Fibre Distributed Data Interface (FDDI) – Token Ring Twisted Pair Physical Layer Medium Dependent (TP-PMD)*

---

IECNORM.COM : Click to view the full PDF of IEC 62439-7:2011

## SOMMAIRE

|                                                                                        |     |
|----------------------------------------------------------------------------------------|-----|
| AVANT-PROPOS .....                                                                     | 86  |
| INTRODUCTION .....                                                                     | 88  |
| 1 Domaine d'application .....                                                          | 90  |
| 2 Références normatives .....                                                          | 90  |
| 3 Termes, définitions, abréviations, acronymes et conventions .....                    | 90  |
| 3.1 Termes et définitions .....                                                        | 90  |
| 3.2 Abréviations et acronymes .....                                                    | 91  |
| 3.3 Conventions .....                                                                  | 92  |
| 4 Présentation d'un RRP .....                                                          | 92  |
| 4.1 Généralités .....                                                                  | 92  |
| 4.2 Contrôle du transfert et de la réception de trames .....                           | 92  |
| 4.2.1 Généralités .....                                                                | 92  |
| 4.2.2 Dispositif normal (ND) et dispositif passerelle (GWD) .....                      | 93  |
| 4.2.3 Comportements du dispositif général (GD) .....                                   | 95  |
| 4.2.4 Comportements du gestionnaire de réseau linéaire (LNM) .....                     | 95  |
| 4.2.5 Comportements des gestionnaires de réseau en anneau (RNM) .....                  | 96  |
| 4.3 Surveillance du statut d'une liaison .....                                         | 96  |
| 4.4 Détection d'erreurs .....                                                          | 97  |
| 4.5 Connexion et activation .....                                                      | 97  |
| 4.6 Prise en charge de la base d'informations pour la gestion de réseau (NMIB) .....   | 98  |
| 4.7 Reprise du réseau .....                                                            | 98  |
| 4.8 Configuration automatique du réseau .....                                          | 98  |
| 4.9 Principe de fonctionnement de base du RRP .....                                    | 98  |
| 5 Comportements de redondance RRP .....                                                | 102 |
| 5.1 Topologie de réseau .....                                                          | 102 |
| 5.2 Reprise dans un réseau en anneau .....                                             | 103 |
| 5.2.1 Généralités .....                                                                | 103 |
| 5.2.2 Anomalie de liaison entre des dispositifs voisins .....                          | 106 |
| 5.2.3 Anomalie de liaison d'un dispositif à distance .....                             | 107 |
| 5.2.4 Anomalie de dispositif sur un RNM .....                                          | 108 |
| 5.3 Procédure d'élection d'un gestionnaire de réseau en anneau (RNM) automatique ..... | 108 |
| 5.3.1 Généralités .....                                                                | 108 |
| 5.3.2 RNM principal (RNMP) .....                                                       | 109 |
| 5.3.3 RNM secondaire (RNMS) .....                                                      | 109 |
| 5.4 Gestion des chemins .....                                                          | 110 |
| 5.4.1 Généralités .....                                                                | 110 |
| 5.4.2 Chemin dans un réseau de topologie linéaire .....                                | 110 |
| 5.4.3 Chemin dans un réseau de topologie en anneau .....                               | 111 |
| 5.5 Collision d'adresses de dispositifs .....                                          | 113 |
| 6 Spécifications des classes RRP .....                                                 | 115 |
| 6.1 Généralités .....                                                                  | 115 |
| 6.2 Modèle .....                                                                       | 115 |
| 6.3 Attributs .....                                                                    | 116 |
| 7 Spécifications des services RRP .....                                                | 122 |
| 7.1 Set device information .....                                                       | 122 |

|       |                                                                                                     |     |
|-------|-----------------------------------------------------------------------------------------------------|-----|
| 7.2   | Get device information .....                                                                        | 124 |
| 7.3   | Get network information .....                                                                       | 126 |
| 7.4   | Get path table information .....                                                                    | 128 |
| 8     | Spécification de protocole RRP .....                                                                | 131 |
| 8.1   | Généralités.....                                                                                    | 131 |
| 8.2   | En-tête Ethernet.....                                                                               | 132 |
| 8.2.1 | Preamble.....                                                                                       | 132 |
| 8.2.2 | Start frame delimiter .....                                                                         | 132 |
| 8.2.3 | Destination MAC address .....                                                                       | 132 |
| 8.2.4 | Source MAC address.....                                                                             | 132 |
| 8.2.5 | Length/Type .....                                                                                   | 132 |
| 8.3   | Codage de RRP_FrameHDR .....                                                                        | 133 |
| 8.3.1 | Version and length .....                                                                            | 133 |
| 8.3.2 | DST_addr.....                                                                                       | 133 |
| 8.3.3 | SRC_addr.....                                                                                       | 134 |
| 8.3.4 | Frame Control (FC) .....                                                                            | 134 |
| 8.4   | Codage de data and pad .....                                                                        | 136 |
| 8.4.1 | Généralités.....                                                                                    | 136 |
| 8.4.2 | Codage de FamilyReq .....                                                                           | 137 |
| 8.4.3 | Codage de FamilyRes .....                                                                           | 137 |
| 8.4.4 | Codage de MediaLinked .....                                                                         | 138 |
| 8.4.5 | Codage de AdvThis .....                                                                             | 138 |
| 8.4.6 | Codage de LineStart.....                                                                            | 139 |
| 8.4.7 | Codage de RingStart .....                                                                           | 140 |
| 8.4.8 | Codage de AckRNMS .....                                                                             | 141 |
| 8.4.9 | Codage de CheckRNMS.....                                                                            | 142 |
| 8.5   | Séquence de contrôle de trame (FCS).....                                                            | 142 |
| 9     | Machine de protocole RRP .....                                                                      | 143 |
| 9.1   | Description du diagramme d'états de protocole .....                                                 | 143 |
| 9.2   | Variables et paramètres locaux pour un état de protocole .....                                      | 144 |
| 9.2.1 | Généralités.....                                                                                    | 144 |
| 9.2.2 | Variables permettant de prendre en charge la gestion des informations de dispositif local.....      | 144 |
| 9.2.3 | Variables permettant de prendre en charge la gestion des informations réseau .....                  | 145 |
| 9.2.4 | Variables permettant de prendre en charge la gestion des informations de chemin de dispositif ..... | 145 |
| 9.2.5 | Variables de la trame RRP reçue.....                                                                | 146 |
| 9.2.6 | Variables locales pour un état de protocole .....                                                   | 146 |
| 9.2.7 | Constantes d'état de protocole .....                                                                | 147 |
| 9.3   | Transitions d'états .....                                                                           | 147 |
| 9.4   | Descriptions des fonctions.....                                                                     | 160 |
| 10    | Base d'informations d'administration RRP (MIB) .....                                                | 166 |
|       | Bibliographie.....                                                                                  | 172 |
|       | Figure 1 – Transfert et réception de trames Ethernet .....                                          | 93  |
|       | Figure 2 – Structures d'un dispositif normal (ND) et d'un dispositif passerelle (GWD) .....         | 94  |
|       | Figure 3 – Contrôle de transfert du LNM .....                                                       | 95  |

|                                                                                                      |     |
|------------------------------------------------------------------------------------------------------|-----|
| Figure 4 – Contrôle de transfert du RNM.....                                                         | 96  |
| Figure 5 – Informations sur le statut de la liaison .....                                            | 97  |
| Figure 6 – Fonctionnement d'un dispositif au cours de la phase d'initialisation.....                 | 99  |
| Figure 7 – Fonctionnement de dispositifs lors de la phase d'établissement d'un réseau linéaire ..... | 99  |
| Figure 8 – Extension du fonctionnement du réseau linéaire .....                                      | 100 |
| Figure 9 – Etablissement d'un réseau en anneau .....                                                 | 101 |
| Figure 10 – Transformation du réseau en anneau en réseau linéaire .....                              | 102 |
| Figure 11 – Topologie d'un réseau en anneau .....                                                    | 103 |
| Figure 12 – Anomalie de liaison entre des dispositifs voisins .....                                  | 106 |
| Figure 13 – Anomalie de liaison d'un dispositif à distance.....                                      | 107 |
| Figure 14 – Anomalie de dispositif sur un RNM.....                                                   | 108 |
| Figure 15 – Gestion des chemins dans un réseau de topologie linéaire .....                           | 110 |
| Figure 16 – Gestion des chemins dans un réseau de topologie en anneau.....                           | 112 |
| Figure 17 – Collision d'adresses de dispositifs RRP dans un réseau en anneau .....                   | 114 |
| Figure 18 – Format de trame MAC courante pour DLPDU RRP.....                                         | 132 |
| Figure 19 – Diagramme d'états de protocole RRP.....                                                  | 143 |
| Tableau 1 – Paramètre de reprise réseau RRP .....                                                    | 104 |
| Tableau 2 – Paramètres de calcul.....                                                                | 104 |
| Tableau 3 – Table de chemins du Dispositif1 dans un réseau de topologie linéaire.....                | 111 |
| Tableau 4 – Table de chemins du Dispositif4 dans un réseau de topologie linéaire.....                | 111 |
| Tableau 5 – Table de chemins du Dispositif1 dans un réseau de topologie en anneau.....               | 112 |
| Tableau 6 – Table de chemins du Dispositif3 dans un réseau de topologie en anneau.....               | 113 |
| Tableau 7 – Informations sur la collision d'adresses de dispositifs .....                            | 114 |
| Tableau 8 – Paramètres du service set device information.....                                        | 123 |
| Tableau 9 – Paramètres du service get device information .....                                       | 125 |
| Tableau 10 – Paramètres du service get network information .....                                     | 127 |
| Tableau 11 – Paramètres du service get path table Information .....                                  | 129 |
| Tableau 12 – Champ Length/Type RRP .....                                                             | 132 |
| Tableau 13 – Version.....                                                                            | 133 |
| Tableau 14 – DST_addr.....                                                                           | 133 |
| Tableau 15 – SRC_addr.....                                                                           | 134 |
| Tableau 16 – Network control message type.....                                                       | 134 |
| Tableau 17 – Type of service.....                                                                    | 135 |
| Tableau 18 – Priority.....                                                                           | 135 |
| Tableau 19 – Validation of extension code.....                                                       | 135 |
| Tableau 20 – Codage de la trame FamilyReq.....                                                       | 137 |
| Tableau 21 – Codage de la trame FamilyRes.....                                                       | 137 |
| Tableau 22 – Codage de la trame MediaLinked .....                                                    | 138 |
| Tableau 23 – Codage de la trame AdvThis.....                                                         | 139 |
| Tableau 24 – Codage de la trame LineStart .....                                                      | 139 |
| Tableau 25 – Codage de la trame RingStart.....                                                       | 140 |

|                                                                                                                     |     |
|---------------------------------------------------------------------------------------------------------------------|-----|
| Tableau 26 – Codage de la trame AckRNMS .....                                                                       | 141 |
| Tableau 27 – Codage de la trame CheckRNMS .....                                                                     | 142 |
| Tableau 28 – Variables permettant de prendre en charge la gestion des informations<br>de dispositif .....           | 145 |
| Tableau 29 – Variables permettant de prendre en charge la gestion des informations<br>réseau .....                  | 145 |
| Tableau 30 – Variables permettant de prendre en charge la gestion des informations<br>de chemin de dispositif ..... | 145 |
| Tableau 31 – Variables de la trame RRP reçue.....                                                                   | 146 |
| Tableau 32 – Variables locales pour un état de protocole .....                                                      | 146 |
| Tableau 33 – Constantes d'état de protocole .....                                                                   | 147 |
| Tableau 34 – Transitions d'états RRP .....                                                                          | 147 |
| Tableau 35 – Descriptions des fonctions RRP.....                                                                    | 160 |

IECNORM.COM : Click to view the full PDF of IEC 62439-7:2011

## COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

### RÉSEAUX DE COMMUNICATION INDUSTRIELS – RÉSEAUX DE HAUTE DISPONIBILITE POUR L'AUTOMATION –

#### Partie 7: Protocole de redondance pour réseau en anneau (RRP)

#### AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de brevet. La CEI 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 CEI 62439-7 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Le texte de cette norme est issu des documents suivants:

| FDIS         | Rapport de vote |
|--------------|-----------------|
| 65C/668/FDIS | 65C/673/RVD     |

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

La présente Norme internationale doit être lue conjointement avec la CEI 62439-1:2010, *Industrial communication networks – High availability automation networks – Part 1: General concepts and calculation methods* (disponible en anglais seulement).

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

Une liste de toutes les parties de la série CEI 62439, présentées sous le titre général *Réseaux de communication industriels – Réseaux de haute disponibilité pour l'automatisation*, est disponible sur le site web de la CEI.

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

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

Le contenu du corrigendum de mai 2015 a été pris en considération dans cet exemplaire.

**IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.**

## INTRODUCTION

La série CEI 62439 spécifie des principes applicables aux réseaux de haute disponibilité répondant aux exigences des réseaux industriels d'automatisation.

Dans un réseau en parfait état de fonctionnement, les protocoles de la série CEI 62439 fournissent à l'ISO/CEI 8802-3:2000 (IEEE 802.3) des communications de données fiables et compatibles, et conservent le déterminisme dans les communications de données en temps réel. En situation de défaillance, de retrait et d'insertion d'un composant, ils indiquent des temps de reprise déterministes.

Ces protocoles conservent en intégralité les capacités de communication Ethernet généralement utilisées dans le domaine tertiaire, et ce, afin de garantir que les logiciels qui dépendent d'eux restent valides.

Le marché recherche différentes solutions de réseau, proposant chacune des caractéristiques de performances et des capacités fonctionnelles différentes, afin de répondre aux nombreuses exigences des applications. Ces solutions prennent en charge des topologies et des mécanismes de redondance différents, qui sont présentés dans la CEI 62439-1 et spécifiés dans les Normes internationales associées. La CEI 62439-1 fait la distinction entre ces solutions pour mieux guider l'utilisateur.

La série CEI 62439 respecte la structure et les conditions générales de la série CEI 61158.

La Commission Electrotechnique Internationale (CEI) attire l'attention sur le fait qu'il est déclaré que la conformité avec les dispositions du présent document peut impliquer l'utilisation d'un brevet intéressant la CEI 61158-4-21 traitée aux Articles 4 et 5.

Numéro de brevet KR 0789444 "COMMUNICATION PACKET PROCESSING APPARATUS AND METHOD FOR RING TOPOLOGY ETHERNET NETWORK CAPABLE OF PREVENTING PERMANENT PACKET LOOPING," détenu par LS INDUSTRIAL SYSTEMS CO., LTD., Anyang, Korea

Numéro de brevet KR 0732510 "NETWORK SYSTEM" détenu par LS INDUSTRIAL SYSTEMS CO., LTD., Anyang, Korea

Numéro de brevet KR 0870670 "Method For Determining a Ring Manager Node", détenu par LS INDUSTRIAL SYSTEMS CO., LTD., Anyang, Korea

La CEI ne prend pas position quant à la preuve, à la validité et à la portée de ces droits de propriété.

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

LSIS Co Ltd  
 LS Tower  
 1026-6, Hogye-Dong  
 Dongan-Gu  
 Anyang, Gyeonggi-Do, 431-848  
 South Korea  
 Téléphone +82 2 2034 4917  
 Fax +82 2 2034 4648

L'attention est d'autre part attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété autres que ceux qui ont été mentionnés ci-dessus. La CEI ne saurait être tenue pour responsable de l'identification de ces droits de propriété en tout ou partie.

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

IECNORM.COM : Click to view the full PDF of IEC 62439-7:2011

## RÉSEAUX DE COMMUNICATION INDUSTRIELS – RÉSEAUX DE HAUTE DISPONIBILITE POUR L'AUTOMATION –

### Partie 7: Protocole de redondance pour réseau en anneau (RRP)

#### 1 Domaine d'application

La série de normes CEI 62439 concerne les réseaux de haute disponibilité pour l'automatisation basés sur la technologie ISO/CEI 8802-3:2000 (Ethernet).

La présente partie de la série de normes CEI 62439 établit les spécifications d'un protocole de redondance basé sur une topologie en anneau, au sein de laquelle il est exécuté aux nœuds d'extrémité, et non pas intégré aux commutateurs. Chaque nœud détecte la défaillance ou l'établissement d'une liaison à l'aide de technologies de détection de supports. Il partage ensuite les informations relatives à la liaison avec les autres nœuds, et ce, dans le but de garantir une reprise rapide de la connectivité. Les nœuds disposent de fonctions de gestion des réseaux à protocole de redondance pour réseau en anneau (RRP).

#### 2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. 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).

CEI 60050-191, *Vocabulaire Electrotechnique International – Chapitre 191: Sûreté de fonctionnement et qualité de service*

CEI 62439-1:2010, *Industrial communication networks – High availability automation networks – Part 1: General concepts and calculation methods* (disponible en anglais uniquement)

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

#### 3 Termes, définitions, abréviations, acronymes et conventions

##### 3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans la CEI 60050-191 et dans la CEI 62439-1 s'appliquent, ainsi que les suivants.

###### 3.1.1

###### **R-port**

port dans un dispositif de communication faisant partie intégrante d'une structure linéaire ou en anneau

###### 3.1.2

###### **device address**

adresse à 2 octets définissant le dispositif associé à un dispositif unique sur une liaison locale spécifique

**3.1.3****Gateway Device****GWD**

dispositif RRP doté de plus de 3 ports Ethernet. Deux ports au moins doivent supporter le protocole RRP

**3.1.4****Normal Device****ND**

dispositif RRP doté de deux ports RRP

**3.1.5****Unique Identification****UID**

identification unique à 8 octets utilisée pour identifier un dispositif RRP au sein d'un segment de réseau. L'UID correspond à la combinaison d'une adresse de dispositif à 2 octets et d'une adresse MAC à 6 octets, permettant ainsi d'obtenir une valeur unique sur le réseau

**3.2 Abréviations et acronymes**

Pour les besoins du présent document, les abréviations et acronymes donnés dans la CEI 62439-1 ainsi que les suivants s'appliquent.

|      |                                                                                              |
|------|----------------------------------------------------------------------------------------------|
| ASE  | Elément de service application ( <i>Application Service Element</i> )                        |
| DLE  | Entité de couche de liaison de données ( <i>Data Link layer Entity</i> )                     |
| FC   | Contrôle de trame ( <i>Frame Control</i> )                                                   |
| FCS  | Séquence de contrôle de trame ( <i>Frame Check Sequence</i> )                                |
| GD   | Dispositif général ( <i>General Device</i> )                                                 |
| GWD  | Dispositif passerelle ( <i>Gateway Device</i> )                                              |
| LNМ  | Gestionnaire de réseau linéaire ( <i>Line Network Manager</i> )                              |
| MAC  | Contrôle d'accès au support ( <i>Media Access Control</i> )                                  |
| MIB  | Base d'informations pour la gestion ( <i>Management Information Base</i> )                   |
| NCM  | Message de contrôle réseau ( <i>Network Control Message</i> )                                |
| NCMT | Type de message de contrôle réseau ( <i>Network Control Message Type</i> )                   |
| ND   | Dispositif normal ( <i>Normal Device</i> )                                                   |
| NMIB | Base d'informations pour la gestion de réseau ( <i>Network Management Information Base</i> ) |
| PHY  | Émetteur-récepteur de l'interface physique                                                   |
| PO   | Sous tension ( <i>Power On</i> )                                                             |
| PRI  | Priorité                                                                                     |
| RES  | Réservé                                                                                      |
| RNM  | Gestionnaire de réseau en anneau ( <i>Ring Network Manager</i> )                             |
| RRP  | Protocole de redondance pour réseau en anneau ( <i>Ring based Redundancy Protocol</i> )      |
| SA   | Autonome ( <i>Stand Alone</i> )                                                              |
| ToS  | Type de service ( <i>Type of Service</i> )                                                   |
| VoE  | Validation du code d'extension ( <i>Validation of Extension code</i> )                       |

### 3.3 Conventions

La présente partie de la série CEI 62439 respecte les conventions définies dans la CEI 62439-1.

## 4 Présentation d'un RRP

### 4.1 Généralités

Le protocole de redondance pour réseau en anneau (RRP) établit les spécifications d'un protocole de reprise, basé sur une topologie en anneau. Toutes les liaisons d'un réseau RRP doivent être "full duplex", grâce à l'utilisation d'un commutateur Ethernet matériel interne. Ainsi, le RRP fournit un mécanisme de transmission sans collision entre deux nœuds. Chaque dispositif RRP détecte la défaillance et l'établissement d'une liaison conformément aux règles spécifiées dans la norme ISO/CEI 8802-3:2000, puis partage cette information avec d'autres dispositifs RRP, et ce, dans le but de garantir une reprise rapide de la connectivité sur le réseau en anneau.

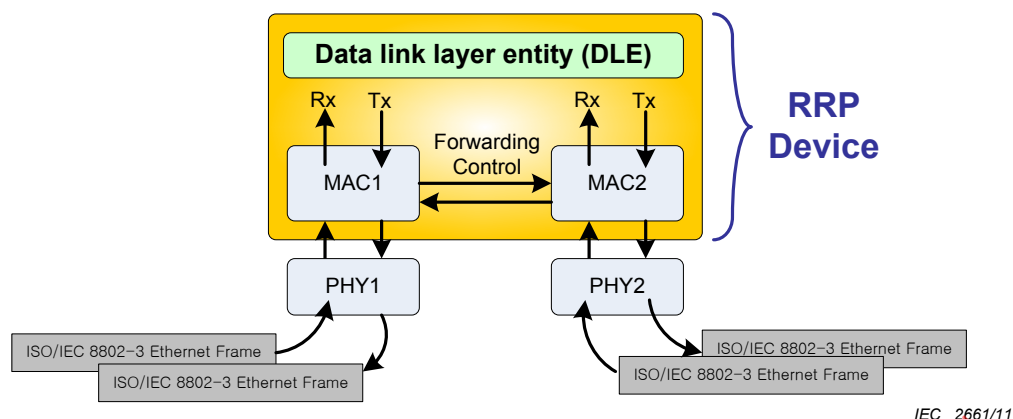
Un dispositif RRP est un dispositif de commutation à deux ports qui reçoit et transmet des trames Ethernet ISO/CEI 8802-3:2000 standard. Il s'agit d'un dispositif intelligent, qui peut contrôler le transfert directionnel de trames entre ses deux ports, en fonction du statut du réseau et de celui du dispositif. Le RRP applique un schéma de gestion de réseau spécial, dont les spécifications sont établies dans la présente Norme. Le RRP applique également un contrôle réseau basé sur les adresses de dispositif et les adresses MAC. Aussi, il est peu probable qu'un concentrateur de pont général ou un commutateur convienne pour un réseau RRP. Cependant, pour connecter un dispositif Ethernet général au réseau RRP, il convient d'utiliser un dispositif passerelle (GWD).

### 4.2 Contrôle du transfert et de la réception de trames

#### 4.2.1 Généralités

Le RRP fournit un mécanisme de transmission sans collision grâce à un commutateur matériel interne "full duplex" avec file d'attente de commutation et double contrôle d'accès au support (MAC) au sein d'un dispositif. La méthode de priorité de commutation entre Tx et le transfert peut être un schéma "Round-Robin", "Tx en premier" ou "Transfert en premier". Toutefois, le RRP ne spécifie aucune méthode de commutation.

Ainsi, un dispositif RRP transmet des trames sans restriction d'accès au support dès qu'elles apparaissent dans la file d'attente de transmission de chaque MAC. La Figure 1 présente le contrôle de transfert et de réception du dispositif RRP.



## Légende

| Anglais                       | Français                                     |
|-------------------------------|----------------------------------------------|
| Data link layer entity (DLE)  | Entité de couche de liaison de données (DLE) |
| Rx                            | Rx                                           |
| Tx                            | Tx                                           |
| MAC1                          | MAC1                                         |
| MAC2                          | MAC2                                         |
| RRP Device                    | Dispositif RRP                               |
| PHY1                          | PHY1                                         |
| PHY2                          | PHY2                                         |
| ISO/IEC 8802-3 Ethernet Frame | Frame Ethernet ISO/CEI 8802-3                |
| Forwarding Control            | Contrôle de transfert                        |

Figure 1 – Transfert et réception de trames Ethernet

## 4.2.2 Dispositif normal (ND) et dispositif passerelle (GWD)

Le RRP est exécuté dans une topologie en anneau à double port. Un dispositif Ethernet général peut envoyer des trames Ethernet standard au travers d'un réseau RRP en anneau avec un dispositif passerelle (GWD). Il est également possible d'établir un réseau à plusieurs anneaux avec un dispositif passerelle (GWD).

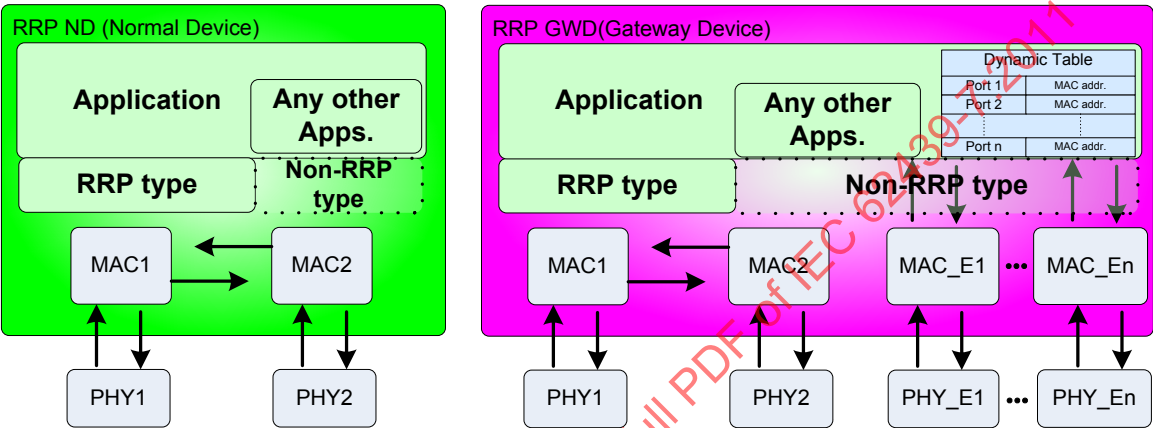
Le dispositif passerelle (GWD) prend en charge la commutation des trames Ethernet entre le réseau RRP et les réseaux Ethernet externes via une couche d'application au moyen d'une table dynamique. La table dynamique répertorie automatiquement les adresses vers les ports Ethernet externes. La table dynamique est automatiquement réalisée d'après les mouvements de trames dans le réseau. Le dispositif passerelle (GWD) inspecte à la fois l'adresse de destination et l'adresse source. L'adresse de destination est utilisée pour la décision de transfert; l'adresse source est utilisée pour ajouter des entrées dans la table et à des fins de mise à jour. Lorsqu'une trame Ethernet est reçue au niveau de la couche du contrôle d'accès au support (MAC) via l'émetteur-récepteur de l'interface physique (PHY), un dispositif passerelle (GWD) prend en charge la trame et exécute l'une des actions suivantes (en fonction de l'adresse MAC et des adresses MAC source de ladite trame):

- pour une trame de diffusion ou multidiffusion, accepter et livrer la trame à l'entité de couche de liaison de données (DLE), et transférer la trame vers l'autre port RRP et les ports Ethernet externes;
- pour une trame affectée au dispositif lui-même, accepter et livrer la trame à la DLE sans transfert;
- pour une trame affectée à l'autre dispositif, accepter la trame à sa couche d'application et inspecter à la fois l'adresse de destination et l'adresse source. Lorsque l'adresse de destination de la trame figure dans la table dynamique, le dispositif passerelle (GWD)

transfère la trame vers le port correspondant dans la table dynamique sans la transférer vers d'autres ports. Dans le cas contraire, le dispositif passerelle (GWD) transfère la trame vers tous les autres ports. Le dispositif passerelle (GWD) ajoute cette entrée dans la table dynamique ainsi que l'adresse MAC source et les informations relatives au numéro de port.

NOTE Les entrées de la table dynamique sont automatiquement supprimées après la durée de vieillissement spécifiée dans l'IEEE 802.1D.

La Figure 2 présente différentes structures d'un dispositif normal (ND) et d'un dispositif passerelle (GWD). Dans le dispositif passerelle (GWD), la connexion Ethernet externe est établie vers le réseau RRP en anneau via MAC\_E et PHY\_E.



IEC 2662/11

Légende

| Anglais                  | Français                        |
|--------------------------|---------------------------------|
| RRP ND (Normal Device)   | RRP ND (dispositif normal)      |
| RRP GWD (Gateway Device) | RRP GWD (dispositif passerelle) |
| Application              | Application                     |
| Any other Apps.          | Toute autre appli.              |
| RRP type                 | Type RRP                        |
| Non-RRP type             | Type non-RRP                    |
| Dynamic Table            | Table dynamique                 |
| Port                     | Port                            |
| MAC addr.                | Adresse Mac                     |
| MAC1                     | MAC1                            |
| MAC2                     | MAC2                            |
| MAC_E1                   | MAC_E1                          |
| MAC_En                   | MAC_En                          |
| PHY1                     | PHY1                            |
| PHY2                     | PHY2                            |
| PHY_E1                   | PHY_E1                          |
| PHY_En                   | PHY_En                          |

Figure 2 – Structures d'un dispositif normal (ND) et d'un dispositif passerelle (GWD)

### 4.2.3 Comportements du dispositif général (GD)

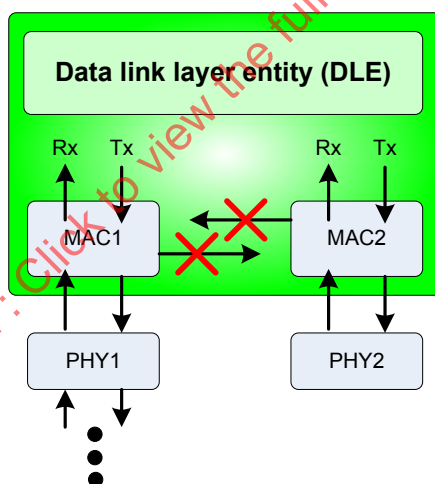
Lorsqu'une trame Ethernet est reçue au niveau de la couche du contrôle d'accès au support (MAC) via l'émetteur-récepteur de l'interface physique (PHY), un dispositif général (GD) RRP autre que le gestionnaire de réseau en anneau (RNM) ou le gestionnaire de réseau linéaire (LNM) prend en charge la trame reçue et exécute l'une des actions suivantes (en fonction de l'adresse MAC et de l'adresse de dispositif de destination de ladite trame):

- pour une trame de diffusion ou multidiffusion, accepter et livrer la trame à l'entité de couche de liaison de données (DLE), et transférer la trame vers l'autre port;
- pour une trame affectée au dispositif lui-même, accepter et livrer la trame à la DLE sans transfert;
- pour une trame affectée à l'autre dispositif, ne pas accepter la trame reçue, mais la transférer vers l'autre port.

Cette procédure de transfert de trames est traitée par le commutateur matériel interne. Elle a donc un impact moindre sur les performances du protocole RRP.

### 4.2.4 Comportements du gestionnaire de réseau linéaire (LNM)

Comme indiqué à la Figure 3, le LNM désactive les fonctions de transfert de trames dans les deux sens. Aussi, les trames ne sont pas transmises vers l'autre port. Sur les réseaux RRP, un LNM est automatiquement configuré. Lorsqu'un dispositif détecte qu'un seul port est connecté, il indique alors qu'il se situe à l'extrémité du réseau linéaire. Le LNM fait également office de point de contrôle du nombre de sauts vers d'autres dispositifs du réseau linéaire.



IEC 2663/11

#### Légende

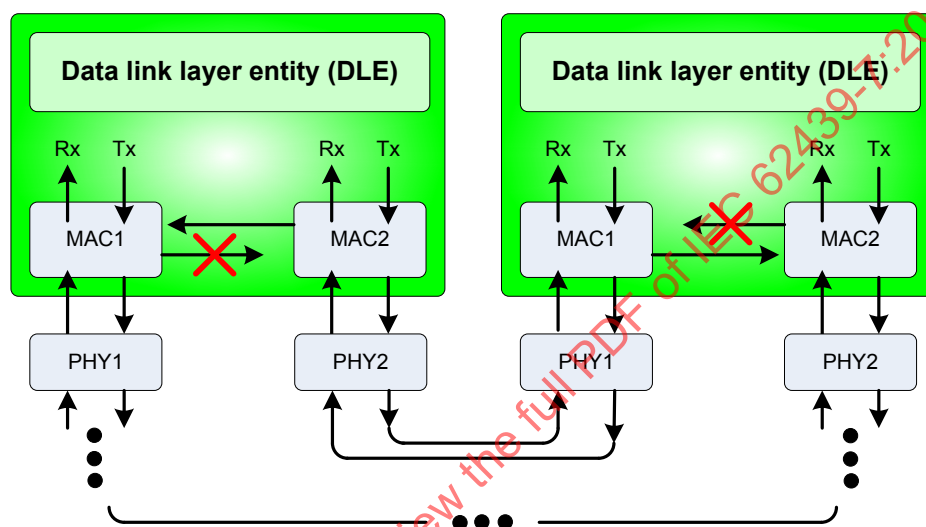
| Anglais                      | Français                                     |
|------------------------------|----------------------------------------------|
| Data link layer entity (DLE) | Entité de couche de liaison de données (DLE) |
| Rx                           | Rx                                           |
| Tx                           | Tx                                           |
| MAC1                         | MAC1                                         |
| MAC2                         | MAC2                                         |
| PHY1                         | PHY1                                         |
| PHY2                         | PHY2                                         |

Figure 3 – Contrôle de transfert du LNM

#### 4.2.5 Comportements des gestionnaires de réseau en anneau (RNM)

Une trame peut circuler de façon continue au sein d'un réseau en anneau lorsque le dispositif affecté est introuvable ou lorsque la trame est diffusée sur le réseau. Dans un réseau en anneau RRP, deux RNM sont automatiquement sélectionnés. Chaque RNM active une seule fonction de transfert directionnel de trames pour éviter une circulation de trame infinie, comme présenté à la Figure 4.

La structure à double RNM permet d'éviter la duplication de message. Un RNM principal (RNMP) est sélectionné tout d'abord en fonction du dispositif ayant l'UID la plus élevée, puis l'un des nœuds voisins est sélectionné en tant que RNM secondaire (RNMS). Le RNMP et le RNMS s'échangent des messages de type contrôle réseau (NCMT) afin de surveiller l'intégrité du réseau.



IEC 2664/11

#### Légende

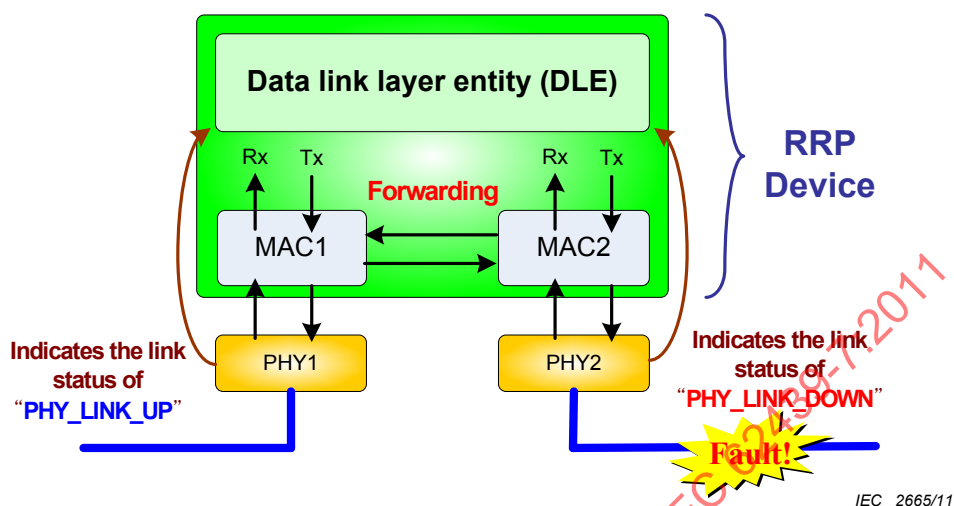
| Anglais                      | Français                                     |
|------------------------------|----------------------------------------------|
| Data link layer entity (DLE) | Entité de couche de liaison de données (DLE) |
| Rx                           | Rx                                           |
| Tx                           | Tx                                           |
| MAC1                         | MAC1                                         |
| MAC2                         | MAC2                                         |
| PHY1                         | PHY1                                         |
| PHY2                         | PHY2                                         |

Figure 4 – Contrôle de transfert du RNM

#### 4.3 Surveillance du statut d'une liaison

Le RRP gère le réseau de façon dynamique. Tout établissement ou interruption de liaison entre deux dispositifs est automatiquement détecté(e) dans la couche physique, comme spécifié dans la norme ISO/CEI 8802-3:2000, Article 24. Les informations relatives au statut de cette liaison sont diffusées et partagées avec tous les dispositifs du réseau grâce à des messages NCMT. Ainsi, la topologie de réseau peut être gérée de façon dynamique. Ces informations indiquent soit "PHY\_LINK\_UP", soit "PHY\_LINK\_DOWN". Le processus de détection du statut de la liaison est lancé par la sous-couche du service PHY. Un statut "PHY\_LINK\_UP" signifie qu'une liaison de communication RRP est établie entre deux dispositifs et qu'il est possible d'envoyer des trames via cette liaison. Un statut "PHY\_LINK\_DOWN" signifie qu'il n'y a aucune liaison de communication RRP via un port

MAC Ethernet et qu'il n'est pas possible d'envoyer des trames par l'intermédiaire du port. Grâce au partage des différentes informations de liaison sur le réseau, tous les dispositifs RRP du réseau ont connaissance du statut de connectivité du réseau en ligne. La Figure 5 présente la procédure de surveillance du statut de la liaison intrinsèque du dispositif RRP.



#### Légende

| Anglais                                      | Français                                     |
|----------------------------------------------|----------------------------------------------|
| Data link layer entity (DLE)                 | Entité de couche de liaison de données (DLE) |
| Rx                                           | Rx                                           |
| Tx                                           | Tx                                           |
| MAC1                                         | MAC1                                         |
| MAC2                                         | MAC2                                         |
| Indicates the link status of "PHY_LINK_UP"   | Indique le statut de liaison "PHY_LINK_UP"   |
| PHY1                                         | PHY1                                         |
| PHY2                                         | PHY2                                         |
| Indicates the link status of "PHY_LINK_DOWN" | Indique le statut de liaison "PHY_LINK_DOWN" |
| Fault!                                       | Anomalie!                                    |
| RRP Device                                   | Dispositif RRP                               |
| Forwarding                                   | Transfert                                    |

Figure 5 – Informations sur le statut de la liaison

#### 4.4 Détection d'erreurs

Un dispositif RRP examine à la fois la validation de trame et le statut de la liaison physique. La validation de trame est étudiée pendant la séquence de contrôle de trame (FCS) spécifiée par la norme ISO/CEI 8802-3:2000, Article 3. Le statut de la liaison physique peut être validé par une fonction de surveillance de liaison PHY. Le RRP utilise un service de la sous-couche PHY pour surveiller le statut de la liaison.

#### 4.5 Connexion et activation

Lorsqu'un nouveau dispositif intègre un réseau existant, les nouvelles informations de liaison sont diffusées via un message NCMT à tous les dispositifs du réseau. Le nouveau dispositif collecte également des informations de liaison existantes à partir de chacun des dispositifs de façon à pouvoir communiquer avec les autres nœuds du réseau sans qu'aucune configuration manuelle ne soit nécessaire.

#### 4.6 Prise en charge de la base d'informations pour la gestion de réseau (NMIB)

Un dispositif RRP gère automatiquement les informations de réseau ainsi qu'une table de chemins. Les informations de réseau et la table de chemins sont enregistrées dans la NMIB du dispositif. Tous les dispositifs RRP d'un réseau partagent des informations de liaison via des messages NCMT. Chaque dispositif met à jour ses informations de réseau et sa table de chemins lorsqu'il reçoit un message NCMT contenant des informations de réseau. Par ailleurs, tous les dispositifs d'un même réseau partagent et collectent des informations de liaison sur le réseau pour mettre à jour celles dont ils disposent et actualiser leur table de chemins. Chaque dispositif met à jour ses informations de réseau et sa table de chemins lorsqu'il reçoit une notification de changement de statut de la liaison.

#### 4.7 Reprise du réseau

En cas de détection d'une défaillance de liaison ou de dispositif sur le réseau en anneau RRP, la topologie est modifiée et les informations relatives au statut de la liaison sont diffusées automatiquement à tous les dispositifs du réseau. Après que les informations relatives à la modification de la topologie ont été diffusées, tous les dispositifs d'un réseau commencent à mettre à jour leur table de chemins et tentent de trouver de nouveaux chemins vers d'autres dispositifs du réseau. Ce processus est exécuté dans la machine de protocole, entre autres pour modifier le point de blocage du réseau. Ainsi, les dispositifs peuvent transmettre des messages à leurs destinataires, tout en mettant à jour leur NMIB.

#### 4.8 Configuration automatique du réseau

Le RRP prend en charge la configuration automatique de réseau. Lorsque la topologie du réseau change, les machines de protocole RRP des dispositifs partagent les informations relatives à la modification du réseau, puis chaque dispositif met à jour sa propre NMIB. Les RNM et les LNM sont automatiquement sélectionnés sur le réseau en fonction de l'UID du dispositif et du statut de connexion.

Lorsqu'un dispositif intègre un réseau RRP, il diffuse un message NCMT contenant les informations de réseau qui le concernent. Les autres dispositifs d'un réseau RRP reçoivent le message NCMT, puis mettent à jour leur NMIB avant de répondre au nouveau dispositif. Ensuite, la NMIB est mise à jour automatiquement au cours du processus d'intégration.

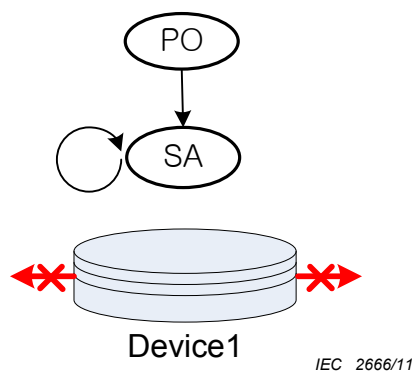
Lorsqu'un dispositif détecte une condition d'anomalie, il diffuse un message NCMT aux autres dispositifs du réseau. Ainsi, les informations de réseau sont gérées automatiquement par les messages NCMT.

#### 4.9 Principe de fonctionnement de base du RRP

Le réseau RRP fonctionne de la façon suivante:

- mise sous tension (initialisation);
- établissement du réseau linéaire;
- extension du réseau linéaire;
- établissement du réseau en anneau;
- modification de la topologie, qui passe d'un réseau en anneau à un réseau linéaire.

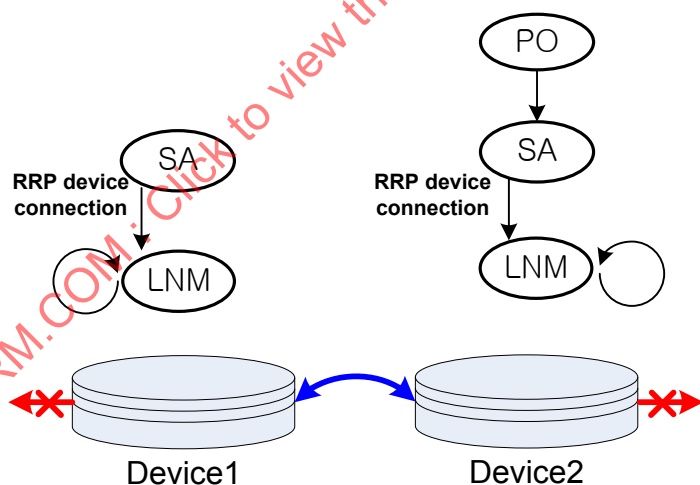
Au cours de la phase d'initialisation, un dispositif RRP devient autonome. Le dispositif tente de trouver des connexions RRP valides sur ses ports. A la Figure 6, le Dispositif1 est initialisé et reste autonome. Au cours de cette phase, le dispositif reste à l'état SA.

**Légende**

| Anglais | Français   |
|---------|------------|
| Device  | Dispositif |
| SA      | SA         |
| PO      | PO         |

**Figure 6 – Fonctionnement d'un dispositif au cours de la phase d'initialisation**

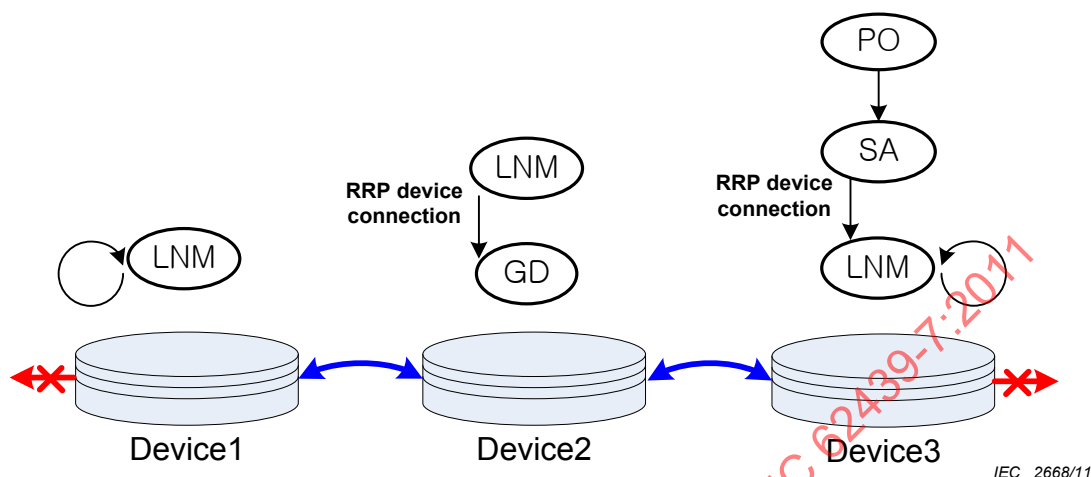
Lorsque le dispositif détecte une connexion RRP valide sur un port, son état passe de SA à LNM. Il diffuse ensuite un message NCMT sur le réseau. La Figure 7 présente la connexion du dispositif et son état.

**Légende**

| Anglais               | Français                 |
|-----------------------|--------------------------|
| Device1               | Dispositif1              |
| Device2               | Dispositif2              |
| SA                    | SA                       |
| PO                    | PO                       |
| RRP device connection | Connexion dispositif RRP |
| LNM                   | LNM                      |

**Figure 7 – Fonctionnement de dispositifs lors de la phase d'établissement d'un réseau linéaire**

Si un ou plusieurs dispositifs sont ajoutés au réseau linéaire, le réseau est étendu et deux dispositifs situés aux extrémités du réseau linéaire conservent la fonction de LNM. Ce processus est exécuté par une machine de protocole RRP dans chaque dispositif ayant une NMIB. La Figure 8 présente une extension du fonctionnement du réseau linéaire.

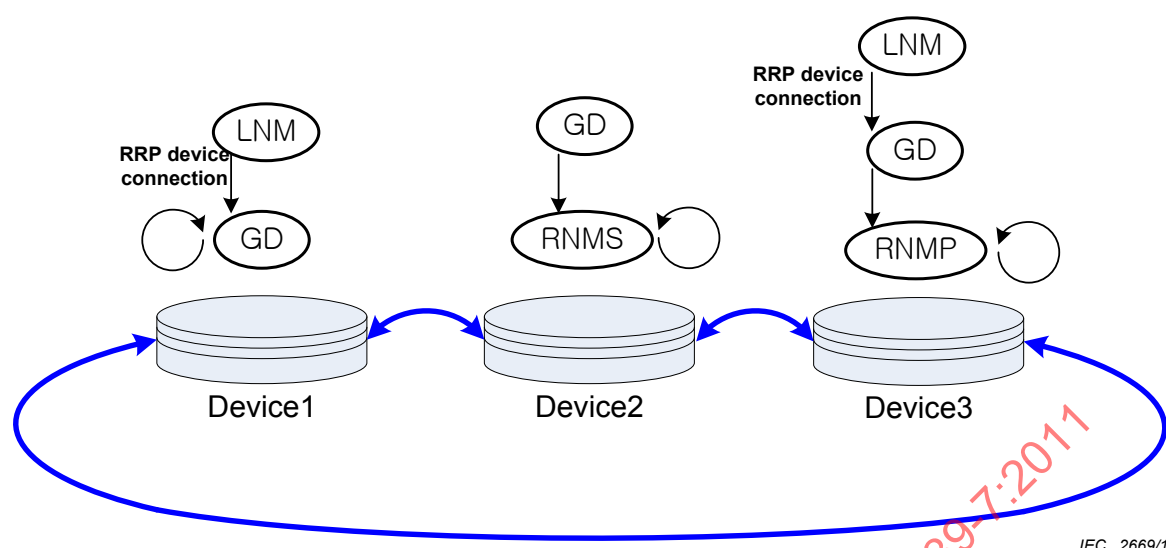


#### Légende

| Anglais               | Français                 |
|-----------------------|--------------------------|
| Device1               | Dispositif1              |
| Device2               | Dispositif2              |
| Device3               | Dispositif3              |
| SA                    | SA                       |
| PO                    | PO                       |
| RRP device connection | Connexion dispositif RRP |
| LNM                   | LNM                      |
| GD                    | GD                       |

**Figure 8 – Extension du fonctionnement du réseau linéaire**

Lorsque les deux extrémités du réseau linéaire sont reliées, le réseau linéaire prend la forme d'un réseau en anneau. Ensuite, le dispositif ayant l'UID la plus élevée envoie un message NCMT en tant que RNMP. Le RNMP choisit le dispositif voisin côté Rport1 comme RNMS. La Figure 9 illustre l'établissement d'un réseau en anneau.



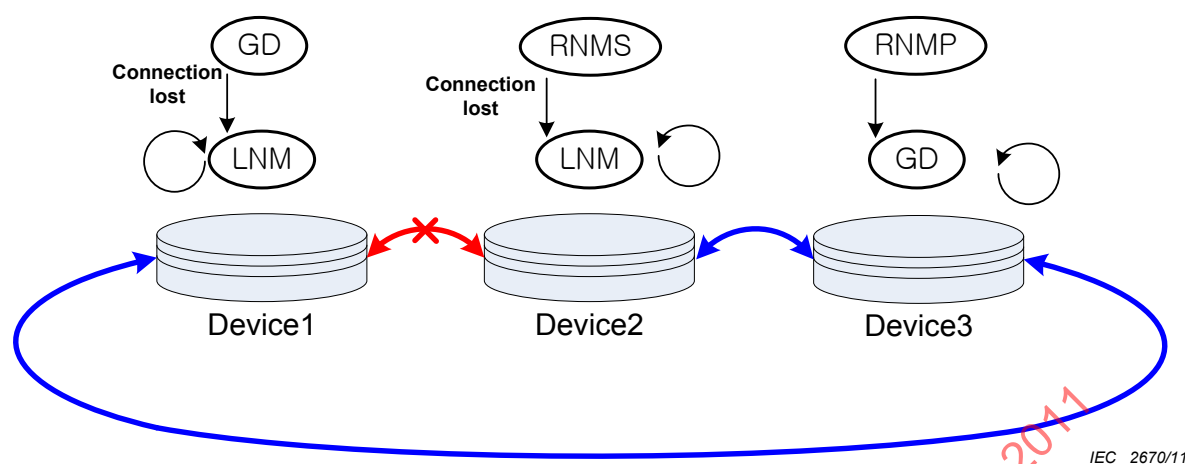
IEC 2669/11

**Légende**

| Anglais               | Français                 |
|-----------------------|--------------------------|
| Device1               | Dispositif1              |
| Device2               | Dispositif2              |
| Device3               | Dispositif3              |
| RNMP                  | RNMP                     |
| PO                    | PO                       |
| GD                    | GD                       |
| RNMS                  | RNMS                     |
| RRP device connection | Connexion dispositif RRP |
| LNM                   | LNM                      |

**Figure 9 – Etablissement d'un réseau en anneau**

Si une condition d'anomalie est détectée, les dispositifs voisins du point d'anomalie diffusent un message NCMT, puis leur état passe à LNM. Le réseau en anneau se transforme alors en réseau linéaire. La Figure 10 présente la transformation de la topologie d'un réseau en anneau en réseau linéaire. Si la perte d'une connexion valide est résolue, le réseau prendra automatiquement la forme d'un réseau en anneau, comme illustré à la Figure 9.



IEC 2670/11

#### Légende

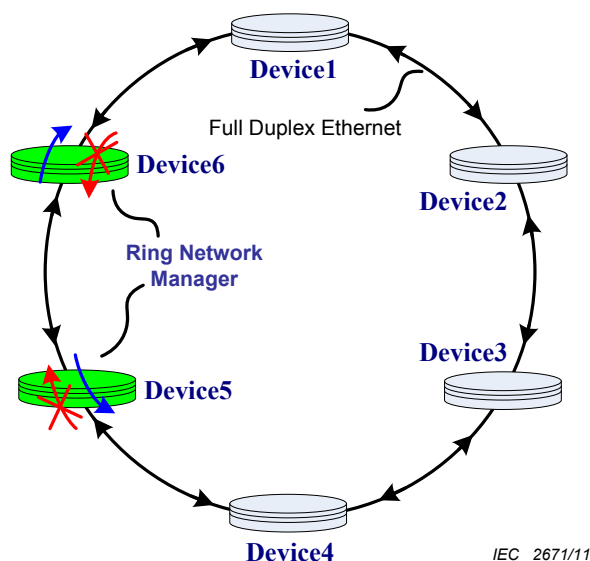
| Anglais         | Français           |
|-----------------|--------------------|
| Device1         | Dispositif1        |
| Device2         | Dispositif2        |
| Device3         | Dispositif3        |
| GD              | GD                 |
| LNM             | LNM                |
| RNMS            | RNMS               |
| RNMP            | RNMP               |
| Connection lost | Perte de connexion |

Figure 10 – Transformation du réseau en anneau en réseau linéaire

## 5 Comportements de redondance RRP

### 5.1 Topologie de réseau

La Figure 11 présente un exemple simple de réseau en anneau RRP.



#### Légende

| Anglais              | Français                         |
|----------------------|----------------------------------|
| Device1              | Dispositif1                      |
| Device2              | Dispositif2                      |
| Device3              | Dispositif3                      |
| Device4              | Dispositif4                      |
| Device5              | Dispositif5                      |
| Device6              | Dispositif6                      |
| Ring Network Manager | Gestionnaire de réseau en anneau |
| Full Duplex Ethernet | Ethernet "Full Duplex"           |

**Figure 11 – Topologie d'un réseau en anneau**

Lorsqu'une nouvelle liaison est établie entre deux nœuds d'extrémité dans un réseau linéaire, le réseau est automatiquement reconfiguré en tant que réseau en anneau. Deux RNM sont automatiquement sélectionnés afin d'éviter la circulation infinie d'une trame au sein d'un réseau en anneau. Toutefois, lorsqu'une défaillance de liaison est détectée sur le réseau en anneau, les gestionnaires de réseau en anneau (RNM) modifient leur état interne afin de transformer leur topologie et passer d'un réseau en anneau à un réseau linéaire. Il serait envisageable que le RNM envoie un message NCMT régulièrement à un autre RNM via le réseau afin d'en contrôler l'intégrité.

## 5.2 Reprise dans un réseau en anneau

### 5.2.1 Généralités

Le processus de reprise du réseau RRP est simple. Il est géré au sein d'une machine de protocole pour garantir une reprise rapide suite aux modifications de topologie de réseau. Lorsqu'une défaillance de liaison est détectée dans un réseau en anneau, le réseau est reconfiguré automatiquement en tant que réseau linéaire. Une modification de topologie impliquant la transformation d'un réseau en anneau en réseau linéaire se déroule dans les trois cas suivants:

- anomalie de la liaison avec le dispositif voisin;
- anomalie de la liaison du dispositif à distance;
- anomalie du dispositif local.

En cas de défaillance de la liaison, les deux dispositifs les plus proches du point d'anomalie prennent l'état LNM et diffusent le message NCMT NCM\_LINE\_START. Lorsqu'un dispositif reçoit NCM\_LINE\_START, il vérifie le statut de sa connexion et prend l'état GD ou LNM.

Une rupture de câble et une interruption d'alimentation sont traitées de la même façon qu'une anomalie de liaison. Une anomalie de dispositif nécessite le même temps de reprise. Le temps de reprise de redondance correspond à l'intervalle qui s'écoule entre le moment où l'anomalie de liaison se produit et celui où le chemin est restauré dans un nouveau sens.

Le RRP spécifie les temps de reprise pour la transition d'un réseau en anneau vers un réseau linéaire. Le Tableau 1 indique le paramètre défini pour le temps de reprise RRP le plus défavorable lorsque le paramètre de réseau est défini comme indiqué dans le Tableau 2.

**Tableau 1 – Paramètre de reprise réseau RRP**

| Paramètre             | 100BASE-X | 1000BASE-X | Définition                                                                                                                                                                                                                                                                        |
|-----------------------|-----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T_fault_sense         | 350 us    | 2 ms       | temps de détection d'anomalie de réseau dans PHY spécifié dans la norme ISO/CEI 8802-3:2000, Article 24, Article 25, Article 26 et Article 36<br>100BASE-TX: ANSI X3.263-1995, Article 10<br>100BASE-FX: ISO/CEI 9314-3:1990, Article 9<br>1000BASE-X: ANSI X3.230-1994, Annexe I |
| T_state_transient     | 1 ms      | 1 ms       | intervalle de temps pour la commutation de la machine de protocole depuis RNM ou GD vers LNM, dans le dispositif RRP                                                                                                                                                              |
| T_message_propagation | 6 ms      | 700 µs     | temps de délivrance du message de NCM_LINE_START depuis un nouveau LNM jusqu'au dispositif le plus éloigné d'un segment                                                                                                                                                           |
| T_recovery (max.)     | ≤ 8 ms    | ≤ 4 ms     | temps de reprise réseau total en ms                                                                                                                                                                                                                                               |

**Tableau 2 – Paramètres de calcul**

| Paramètre | Valeur     |            | Définition                                                            |
|-----------|------------|------------|-----------------------------------------------------------------------|
|           | 100BASE-X  | 1000BASE-X |                                                                       |
| $N$       | 50         | 50         | nombre de nœuds compris entre les dispositifs d'envoi et de réception |
| $T_{PKT}$ | 24 µs      | 2,4 µs     | temps de transmission de paquets                                      |
| NCMsize   | 112 octets | 112 octets | longueur du message NCM_LINE_START                                    |
| POsize    | 40 octets  | 40 octets  | volume de traitement des protocoles                                   |
| LDR       | 100        | 1 000      | vitesse de la liaison en Mbit/s                                       |
| $T_{CPD}$ | 0,5 µs     | 0,05 µs    | temps de propagation de câble du nœud pour 100 m                      |
| $T_{SND}$ | 50 µs      | 50 µs      | temps de traversée de la pile de l'expéditeur, y compris PHY et MAC   |
| $T_{RCV}$ | 50 µs      | 50 µs      | temps de traversée de la pile du destinataire, y compris PHY et MAC   |
| $T_{NLD}$ | 120 µs     | 12 µs      | temporisation de la latence du nœud dans le cas le plus défavorable   |
|           | 3 µs       | 0,3 µs     | temporisation de la latence du nœud dans le cas le plus favorable     |

Le temps de reprise du réseau peut être calculé par l'Equation (1)

$$T_{\text{RECOVER}} = T_{\text{FS}} + T_{\text{RtoL}} + T_{\text{MSG}} \quad (1)$$

où

- $T_{\text{RECOVER}}$  est le temps de reprise du réseau en ms;  
 $T_{\text{FS}}$  est le temps de détection d'anomalie de la liaison en ms;  
 $T_{\text{RtoL}}$  est la temporisation de la transition d'état du dispositif en  $\mu\text{s}$ ;  
 $T_{\text{MSG}}$  est le temps de propagation du message NCM\_LINE\_START jusqu'au dispositif le plus éloigné en ms, voir l'Equation (2).

Le temps de propagation du message NCMT, NCM\_LINE\_START, jusqu'au dispositif le plus éloigné  $T_{\text{MSG}}$  peut être calculé par l'Equation (2).

$$T_{\text{MSG}} = T_{\text{SND}} + T_{\text{PKT}} + T_{\text{CPD}} + \sum_{i=0}^N T_{\text{NLD}_i} + T_{\text{RCV}} \quad (2)$$

où

- $T_{\text{MSG}}$  est le temps de propagation du message en  $\mu\text{s}$ ;  
 $T_{\text{SND}}$  est le temps de traversée de la pile de l'expéditeur, y compris PHY et MAC, en  $\mu\text{s}$ ;  
 $T_{\text{PKT}}$  est le temps de transmission de paquets en  $\mu\text{s}$ , voir l'Equation (3);  
 $T_{\text{CPD}}$  est le temps de propagation du câble pour un nœud, en  $\mu\text{s}$   
 $T_{\text{NLD}_i}$  est la temporisation de latence d'un nœud  $i$  en microsecondes, voir l'Equation (4);  
 $T_{\text{RCV}}$  est le temps de traversée de la pile du destinataire, y compris PHY et MAC, en  $\mu\text{s}$ ;  
 $N$  est le nombre de nœuds compris entre les dispositifs d'envoi et de réception.

Le temps de transmission de paquets  $T_{\text{PKT}}$  peut être calculé par l'Equation (3)

$$T_{\text{PKT}} = \frac{(\text{NCMsize} + \text{POsize}) \times 8}{\text{LDR}} \quad (3)$$

où

- $T_{\text{PKT}}$  est le temps de transmission de paquets sur le support, en  $\mu\text{s}$ ;  
 $\text{NCMsize}$  est la taille de l'unité de données NCM en octets;  
 $\text{LDR}$  est le débit de données de liaison en bits par seconde;  
 $\text{POsize}$  est le volume de traitement des protocoles en octets.

La temporisation de la latence du nœud  $i$   $T_{\text{NLD}_i}$  peut être calculée par l'Equation (4)

$$T_{\text{NLD}_i} = T_{\text{NPD}_i} + T_{\text{PKT}_i} + \sum_{j=0}^M T_{\text{TX\_PKT}_{ij}} \quad (4)$$

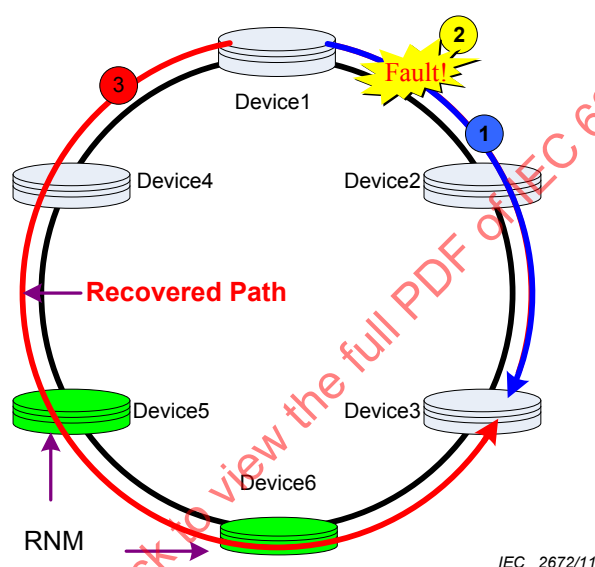
où

- $T_{\text{NLD}_i}$  est la temporisation de latence du nœud  $i$  en  $\mu\text{s}$ ;

- $T_{NPD\_i}$  est la temporisation de propagation du nœud  $i$  en  $\mu s$ ;
- $T_{PKT\_i}$  est le temps de transmission de paquets du nœud  $i$  en  $\mu s$ ;
- $T_{TX\_PKT\_ij}$  est le temps en ms de transmission de paquets précédant la transmission du paquet  $j$  dans la file d'attente de transmission au port du nœud  $i$ ;
- $M$  est le nombre de paquets situés avant ce paquet dans la file d'attente de transmission au port du nœud  $i$ .

### 5.2.2 Anomalie de liaison entre des dispositifs voisins

La Figure 12 présente un exemple d'anomalie de liaison entre des dispositifs voisins dans un réseau en anneau.



#### Légende

| Anglais        | Français        |
|----------------|-----------------|
| Device1        | Dispositif1     |
| Device2        | Dispositif2     |
| Device3        | Dispositif3     |
| Device4        | Dispositif4     |
| Device5        | Dispositif5     |
| Device6        | Dispositif6     |
| Recovered Path | Chemin restauré |
| RNM            | RNM             |
| Fault!         | Anomalie!       |

**Figure 12 – Anomalie de liaison entre des dispositifs voisins**

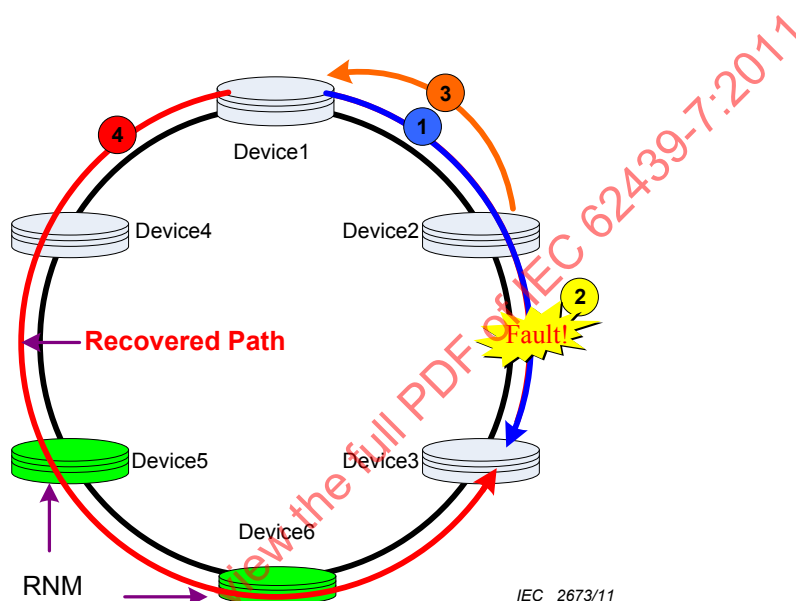
Si la liaison entre le Dispositif1 et le Dispositif2 est interrompue alors que le Dispositif1 tente d'envoyer une trame au Dispositif3, un événement d'anomalie de liaison est spontanément déclenché par un signal matériel. La machine de protocole RRP le détecte dans le Dispositif1. Le Dispositif1 décide par la suite que la liaison est indisponible pour la transmission de données. Il convient alors de reconfigurer le réseau en anneau sous forme de réseau linéaire, en définissant l'état du Dispositif1 sur LNM. Le Dispositif1 diffuse un message NCM\_LINE\_START et modifie le R-port de destination par Dispositif3. Le Dispositif1 transmet

la trame au nouveau R-port de destination. Le Dispositif2 diffuse également un message NCM\_LINE\_START pour déclarer l'anomalie de liaison et définit son propre état sur LNM.

Dans le cas où le Dispositif2 ne parviendrait pas à détecter l'anomalie de liaison, le Dispositif5 reçoit le message NCM\_LINE\_START du Dispositif1 et passe de l'état RNM à GD. Puis, le Dispositif5 transmet le message NCM\_LINE\_START au Dispositif6. Ainsi, d'autres dispositifs du réseau reconnaissent la modification de topologie.

### 5.2.3 Anomalie de liaison d'un dispositif à distance

La Figure 13 présente un exemple d'anomalie de liaison d'un dispositif à distance dans un réseau en anneau.



#### Légende

| Anglais        | Français        |
|----------------|-----------------|
| Device1        | Dispositif1     |
| Device2        | Dispositif2     |
| Device3        | Dispositif3     |
| Device4        | Dispositif4     |
| Device5        | Dispositif5     |
| Device6        | Dispositif6     |
| Recovered Path | Chemin restauré |
| RNM            | RNM             |
| Fault!         | Anomalie!       |

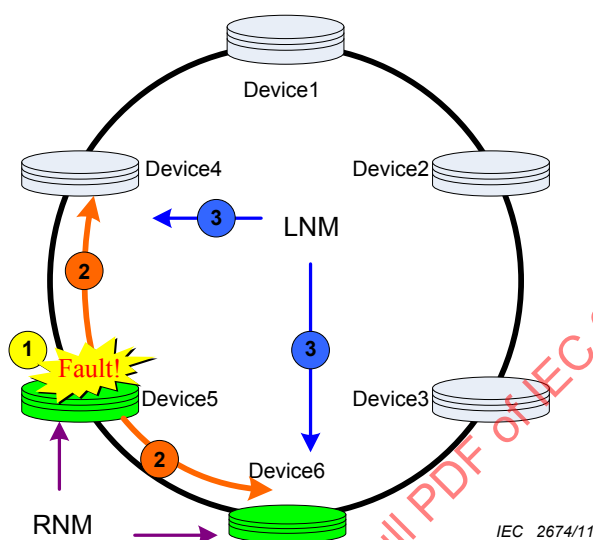
**Figure 13 – Anomalie de liaison d'un dispositif à distance**

Si la liaison entre le Dispositif2 et le Dispositif3 est interrompue alors que le Dispositif1 tente d'envoyer une trame au Dispositif3, un événement d'anomalie de liaison est spontanément déclenché par un signal matériel. La machine de protocole RRP le détecte dans le Dispositif2. A ce stade, le Dispositif2 diffuse un message NCM\_LINE\_START indiquant que la liaison n'est pas disponible et qu'il convient de reconfigurer le réseau en anneau sous la forme d'un réseau linéaire. Le Dispositif1 modifie le R-port de destination en Dispositif3, qui fait office de deuxième R-port, et transmet la trame via le nouveau R-port. Le Dispositif3 diffuse également

un message NCM\_LINE\_START pour déclarer l'anomalie de liaison et définit son propre état sur LNM.

#### 5.2.4 Anomalie de dispositif sur un RNM

La Figure 14 présente un exemple d'anomalie de dispositif sur un RNM dans un réseau en anneau.



#### Légende

| Anglais | Français    |
|---------|-------------|
| Device1 | Dispositif1 |
| Device2 | Dispositif2 |
| Device3 | Dispositif3 |
| Device4 | Dispositif4 |
| Device5 | Dispositif5 |
| Device6 | Dispositif6 |
| LNM     | LNM         |
| RNM     | RNM         |
| Fault!  | Anomalie!   |

**Figure 14 – Anomalie de dispositif sur un RNM**

Si une anomalie se produit sur le Dispositif5, le Dispositif6 et le Dispositif4 détectent une anomalie de liaison sur leurs R-ports communs au Dispositif5. Le Dispositif6 et le Dispositif4 diffusent tous deux des messages NCM\_LINE\_START à d'autres dispositifs et définissent leur état sur LNM. Une anomalie de GD est traitée de façon identique à une anomalie de RNM.

### 5.3 Procédure d'élection d'un gestionnaire de réseau en anneau (RNM) automatique

#### 5.3.1 Généralités

Pour éviter une circulation de trame infinie dans un réseau en anneau RRP, le RNM principal (RNMP) et le RNM secondaire (RNMS) sont sélectionnés automatiquement à partir de leurs identifications uniques (UID) de dispositif, de la façon suivante:

- lorsque deux LNM sont connectés, chaque LNM envoie un message NCMT par diffusion ou multidiffusion;
- chaque dispositif qui reçoit un message NCMT répond avec son UID par diffusion ou multidiffusion;
- lorsqu'un dispositif reçoit une réponse à son propre message NCMT, il détecte que le réseau est un réseau en anneau et, par conséquent, ne transfère pas le message;
- le dispositif ayant l'UID la plus élevée au sein du réseau en anneau est automatiquement sélectionné en tant que RNMP;
- le RNMP envoie un message NCM\_RING\_START, comprenant notamment la demande d'affectation de RNMS, au dispositif voisin connecté via le R-port1 et le R-port2;
- le RNMS répond par un message NCM\_ACK\_RNMS au RNMP;
- la configuration automatique de réseau en anneau est exécutée par le RNMP et le RNMS;
- le RNMP bloque le transfert de trame vers le R-port côté RNMS, tandis que le RNMS bloque également le transfert de trame vers le R-port côté RNMP afin d'éviter une circulation en boucle sur le réseau en anneau.

### 5.3.2 RNM principal (RNMP)

Un dispositif RRP comprend que le réseau a une topologie en anneau lorsque le message NCMT qu'il a lui-même généré lui revient par l'autre port. Dans un réseau en anneau, le dispositif ayant l'UID la plus élevée est sélectionné en tant que RNMP. Lorsqu'un dispositif RRP détecte que le réseau est configuré sous forme d'anneau, tous les dispositifs tentent de trouver celui qui a l'UID la plus élevée dans la table de chemins. Aussi, la mise en concurrence concernant la sélection du RNMP n'est pas nécessaire dans le RRP. Si, sur le réseau, le dispositif ayant l'UID la plus élevée est un dispositif à distance, les autres dispositifs attendent qu'il envoie un message NCM\_RING\_START. Si, sur le réseau, le dispositif ayant l'UID la plus élevée est un dispositif local, il est alors sélectionné en tant que RNMP. La machine de protocole RRP désactive donc les deux fonctions de transfert de trames pour éviter toute circulation en boucle sur le réseau. Elle génère un message NCM\_RING\_START comprenant notamment les informations RNMS relatives au dispositif connecté via le R-port1 et le R-port2 du RNMP.

Lorsque le RNMP reçoit le message NCM\_ACK\_RNMS depuis le RNMS, il désactive la fonction de transfert de trames dans le sens RNMS, mais laisse l'autre sens activé. Le RNMS désactive la fonction de transfert de trames dans le sens RNMP et l'active dans l'autre sens.

Il serait envisageable que le RNMP envoie régulièrement un message NCM\_CHECK\_RNMS au RNMS comme trame de vérification de l'intégrité du réseau. Si la réponse au message NCM\_CHECK\_RNMS n'est pas parvenue dans l'intervalle de temps RRP\_ChkRNMS depuis le RNMS et que le nombre de tentatives de vérification d'intégrité du réseau est dépassé, le RNMP considère qu'une erreur s'est produite sur le réseau.

NOTE 1 L'UID de dispositif a une valeur unique sur le réseau. Le RNMP et le RNMS sont donc sélectionnés automatiquement, et ce, même dans le cadre d'une collision d'adresses de dispositifs (voir 5.5).

NOTE 2 Les messages NCM\_CHECK\_RNMS et NCM\_ACK\_RNMS permettent de sélectionner le RNMS et de le faire valider par le RNMP. Par ailleurs, le RRP prend en charge ces messages, lesquels sont susceptibles de permettre à l'utilisateur de vérifier l'intégrité du réseau. L'intervalle de temps prévu et le nombre de tentatives possibles sont indiqués sous la forme de variables locales (voir Tableau 32).

### 5.3.3 RNM secondaire (RNMS)

Le RNMS est affecté par le RNMP. Lorsqu'un dispositif GD reçoit un message NCM\_RING\_START depuis le RNMP, la machine de protocole RRP compare l'UID du dispositif local à celle du dispositif RNMS, lesquelles figurent dans le message. Si le dispositif n'est pas affecté en tant que RNMS, la machine de protocole RRP active les deux fonctions de transfert de trames dans le dispositif à l'état GD. Si le dispositif est affecté en tant que RNMS, la machine de protocole RRP prend l'état RNMS et transmet NCM\_ACK\_RNMS au RNMP via le R-port reçu du message NCM\_RING\_START, ainsi que le message

NCM\_CHECK\_RNMS depuis le RNMP. Par ailleurs, le dispositif RNMS désactive la fonction de transfert de trames dans le sens RNMP, mais l'active dans l'autre sens.

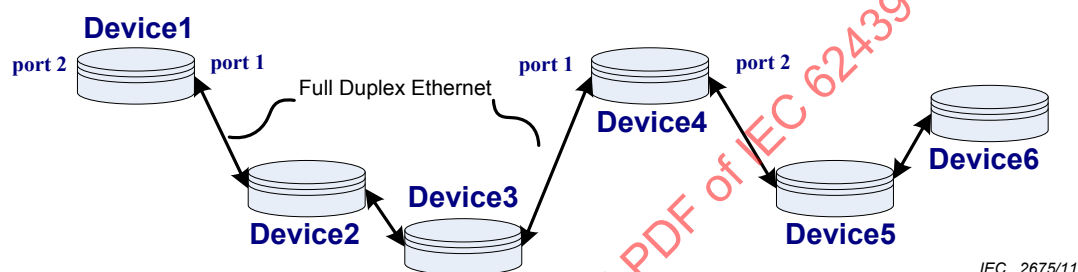
## 5.4 Gestion des chemins

### 5.4.1 Généralités

Le dispositif RRP fournit des informations sur les chemins pour tous les dispositifs RRP du réseau. La gestion des chemins est évaluée à partir du nombre de sauts. Le nombre de sauts indique le nombre de transferts de trames requis pour transmettre une trame au dispositif de destination.

### 5.4.2 Chemin dans un réseau de topologie linéaire

Dans un réseau linéaire, un chemin unique conduit au dispositif de destination. La Figure 15 présente un exemple de gestion des chemins dans un réseau de topologie linéaire.



#### Légende

| Anglais              | Français               |
|----------------------|------------------------|
| Device1              | Dispositif1            |
| Device2              | Dispositif2            |
| Device3              | Dispositif3            |
| Device4              | Dispositif4            |
| Device5              | Dispositif5            |
| Device6              | Dispositif6            |
| Port 1               | Port 1                 |
| Port 2               | Port 2                 |
| Full Duplex Ethernet | Ethernet "Full Duplex" |

**Figure 15 – Gestion des chemins dans un réseau de topologie linéaire**

Le Tableau 3 indique la table de chemins du Dispositif1 et le Tableau 4, celui du Dispositif4, à la Figure 15. Dans un réseau linéaire, un chemin unique est possible entre deux dispositifs, quels qu'ils soient.

**Tableau 3 – Table de chemins du Dispositif1  
dans un réseau de topologie linéaire**

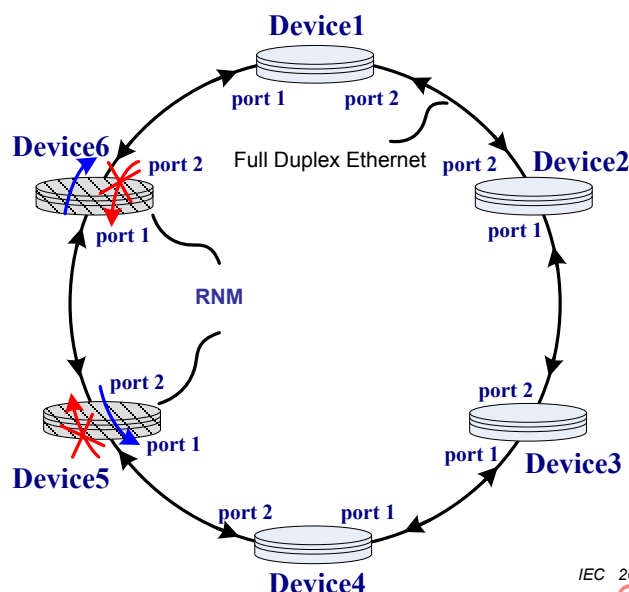
| R-port              | Destination |             |             |             |             |
|---------------------|-------------|-------------|-------------|-------------|-------------|
|                     | Dispositif2 | Dispositif3 | Dispositif4 | Dispositif5 | Dispositif6 |
| R-port1             | 0 saut      | 1 saut      | 2 sauts     | 3 sauts     | 4 sauts     |
| R-port2             | Non valide  | Non valide  | Non valide  | Non valide  | Non valide  |
| Port préférentiel   | R-port1     | R-port1     | R-port1     | R-port1     | R-port1     |
| Port de destination | R-port1     | R-port1     | R-port1     | R-port1     | R-port1     |

**Tableau 4 – Table de chemins du Dispositif4  
dans un réseau de topologie linéaire**

| R-port              | Destination |             |             |             |             |
|---------------------|-------------|-------------|-------------|-------------|-------------|
|                     | Dispositif1 | Dispositif2 | Dispositif3 | Dispositif5 | Dispositif6 |
| R-port1             | 2 sauts     | 1 saut      | 0 saut      | Non valide  | Non valide  |
| R-port2             | Non valide  | Non valide  | Non valide  | 0 saut      | 1 saut      |
| Port préférentiel   | R-port1     | R-port1     | R-port1     | R-port2     | R-port2     |
| Port de destination | R-port1     | R-port1     | R-port1     | R-port2     | R-port2     |

#### 5.4.3 Chemin dans un réseau de topologie en anneau

Dans un réseau en anneau, deux chemins sont possibles entre deux dispositifs, quels qu'ils soient: le chemin dans le sens horaire et le chemin dans le sens antihoraire. Toutefois, une trame ne peut pas être transférée sur le RNMP ou le RNMS. Aussi, il n'est pas possible de transférer une trame par un chemin qui passe par le RNMP ou le RNMS. La Figure 16 présente un exemple de gestion des chemins dans un réseau en anneau.



**Légende**

| Anglais              | Français               |
|----------------------|------------------------|
| Device1              | Dispositif1            |
| Device2              | Dispositif2            |
| Device3              | Dispositif3            |
| Device4              | Dispositif4            |
| Device5              | Dispositif5            |
| Device6              | Dispositif6            |
| Full Duplex Ethernet | Ethernet "Full Duplex" |
| port                 | port                   |
| RNM                  | RNM                    |

**Figure 16 – Gestion des chemins dans un réseau de topologie en anneau**

Le Tableau 5 illustre la table de chemins du Dispositif1 à la Figure 16. Le chemin le plus court pour aller du Dispositif1 au Dispositif5 va dans le sens R-port1. Or, ce chemin est bloqué par le RNM (Dispositif6). Dans le cas présent, le chemin de destination correspond au sens R-port2.

**Tableau 5 – Table de chemins du Dispositif1 dans un réseau de topologie en anneau**

| R-port              | Destination |             |               |                                                     |                   |
|---------------------|-------------|-------------|---------------|-----------------------------------------------------|-------------------|
|                     | Dispositif2 | Dispositif3 | Dispositif4   | Dispositif5 (RNM)                                   | Dispositif6 (RNM) |
| R-port1             | 4 sauts     | 3 sauts     | 2 sauts       | 1 saut                                              | 0 saut            |
| R-port2             | 0 saut      | 1 saut      | 2 sauts       | 3 sauts                                             | 4 sauts           |
| Port préférentiel   | R-port2     | R-port2     | Peu importe   | R-port1<br>(Sens R-port1 bloqué par le Dispositif6) | R-port1           |
| Port de destination | R-port2     | R-port2     | R-port1(NOTE) | R-port2                                             | R-port1           |

NOTE Si les deux chemins ont le même nombre de sauts et qu'ils ne sont pas bloqués par les RNM, le sens R-port1 est toujours privilégié.

Le Tableau 6 présente la table de chemins du Dispositif3 à la Figure 16.

**Tableau 6 – Table de chemins du Dispositif3  
dans un réseau de topologie en anneau**

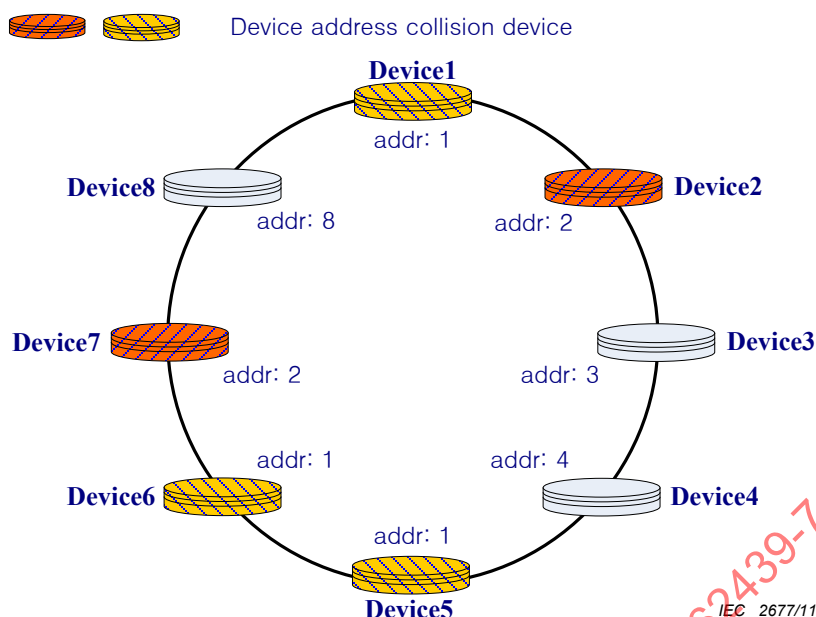
| R-port              | Destination |             |             |                   |                                                            |
|---------------------|-------------|-------------|-------------|-------------------|------------------------------------------------------------|
|                     | Dispositif1 | Dispositif2 | Dispositif4 | Dispositif5 (RNM) | Dispositif6 (RNM)                                          |
| R-port1             | 3 sauts     | 4 sauts     | 0 saut      | 1 saut            | 2 sauts                                                    |
| R-port2             | 1 saut      | 0 saut      | 4 sauts     | 3 sauts           | 2 sauts                                                    |
| Port préférentiel   | R-port2     | R-port2     | R-port1     | R-port1           | Peu importe<br>(Sens R-port1 bloqué<br>par le Dispositif5) |
| Port de destination | R-port2     | R-port2     | R-port1     | R-port1           | R-port2                                                    |

### 5.5 Collision d'adresses de dispositifs

Une adresse de dispositif RRP est configurée manuellement à l'aide de paramètres matériels (p. ex., interrupteur rotatif) ou définie de façon logicielle (p. ex., terminal de données portatif). Il est permis de dupliquer l'adresse de dispositif dans d'autres dispositifs du réseau en cas d'erreur de configuration. Lorsque cette situation se produit, le dispositif n'est plus en mesure de communiquer avec d'autres dispositifs. Ce type d'événement de collision d'adresses de dispositifs sur le réseau est automatiquement détecté par la machine de protocole RRP.

Le RRP utilise une UID à 8 octets pour gérer les ressources réseau, en lieu et place d'une adresse de dispositif. L'UID est créée par combinaison d'une adresse de dispositif et d'une adresse MAC, afin d'obtenir une valeur unique sur le réseau. Les informations relatives au réseau et à l'UID sont stockées dans la NMIB de chaque dispositif. Ainsi, les dispositifs du réseau pourraient avoir la possibilité de déterminer ceux qui disposent d'une adresse dupliquée.

La Figure 17 présente un exemple de collision d'adresses de dispositifs dans un réseau en anneau.



**Légende**

| Anglais                         | Français                           |
|---------------------------------|------------------------------------|
| Device1                         | Dispositif1                        |
| Device2                         | Dispositif2                        |
| Device3                         | Dispositif3                        |
| Device4                         | Dispositif4                        |
| Device5                         | Dispositif5                        |
| Device6                         | Dispositif6                        |
| Device7                         | Dispositif7                        |
| Device8                         | Dispositif8                        |
| addr:                           | adr.:                              |
| Device address collision device | Dispositif en collision d'adresses |

**Figure 17 – Collision d'adresses de dispositifs RRP dans un réseau en anneau**

Les dispositifs 1, 5 et 6 ont la même adresse, d'une valeur de 1. Le Dispositif2 et le Dispositif7 ont eux aussi la même adresse, d'une valeur de 2. Par conséquent, trois événements de collision d'adresses de dispositifs sont détectés par la machine de protocole RRP. Ces événements sont recensés par des compteurs de collision dans la base d'informations pour la gestion de réseau, puis partagés par tous les dispositifs du réseau.

Le Tableau 7 affiche les informations relatives à la collision d'adresses de dispositifs pour les situations de la Figure 17 qui représentent le mécanisme de détection de collision à l'aide des UID de dispositif.

**Tableau 7 – Informations sur la collision d'adresses de dispositifs**

| Nom de dispositif | Adresse de dispositif | Adresse MAC    | UID de dispositif  | Collision |
|-------------------|-----------------------|----------------|--------------------|-----------|
| Dispositif1       | 0x0001                | 0x002233445511 | 0x0001002233445511 | Collision |
| Dispositif2       | 0x0002                | 0x002233445522 | 0x0002002233445522 | Collision |
| Dispositif3       | 0x0003                | 0x002233445533 | 0x0003002233445533 | —         |
| Dispositif4       | 0x0004                | 0x002233445544 | 0x0004002233445544 | —         |

| Nom de dispositif | Adresse de dispositif | Adresse MAC    | UID de dispositif  | Collision |
|-------------------|-----------------------|----------------|--------------------|-----------|
| Dispositif5       | 0x0001                | 0x002233445555 | 0x0001002233445555 | Collision |
| Dispositif6       | 0x0001                | 0x002233445566 | 0x0001002233445566 | Collision |
| Dispositif7       | 0x0002                | 0x002233445577 | 0x0002002233445577 | Collision |
| Dispositif8       | 0x0008                | 0x002233445588 | 0x0008002233445588 | —         |

L'adresse de dispositif est configurée manuellement par l'opérateur. Sa valeur est comprise entre 0-255. L'adresse MAC est une adresse MAC Ethernet ISO/CEI 8802-3:2000 unique à 6 octets. L'UID du dispositif RRP contient une adresse de dispositif RRP à 2 octets et une adresse MAC à 6 octets. Par conséquent, l'UID du dispositif RRP possède une valeur unique de 8 octets sur le réseau qui ne peut être dupliquée. Cette UID permet d'identifier un dispositif spécifique sur le réseau.

Une collision d'adresses de dispositifs RRP est détectée par la machine de protocole RRP lorsque des dispositifs ayant des UID différentes sont configurés avec la même adresse. Si une collision d'adresses de dispositifs locaux est détectée, le dispositif définit l'indicateur de collision d'adresses parmi les indicateurs des dispositifs locaux et génère un événement de collision. Si une collision d'adresses de dispositifs à distance est détectée, le dispositif définit l'indicateur de collision d'adresses parmi les indicateurs du réseau et génère un événement de collision.

## 6 Spécifications des classes RRP

### 6.1 Généralités

L'élément de service d'application (ASE) RRP définit un type d'objet.

### 6.2 Modèle

Un objet RRP se décrit sous le modèle suivant:

**ASE:** ASE de redondance basée sur une structure en anneau  
**CLASSE:** Redondance basée sur une structure en anneau  
**ID CLASSE:** non utilisé  
**CLASSE PARENT:** IEEE 802.3 Protocole de redondance pour réseau en anneau  
**ATTRIBUTS:**

|    |     |                     |                                  |
|----|-----|---------------------|----------------------------------|
| 1  | (m) | Attribut principal: | Local device address             |
| 2  | (m) | Attribut:           | Local device flags               |
| 3  | (m) | Attribut:           | Local device state               |
| 4  | (m) | Attribut:           | Local device unique ID           |
| 5  | (m) | Attribut:           | Device UID for R-port1           |
| 6  | (m) | Attribut:           | Device UID for R-port2           |
| 7  | (m) | Attribut:           | Local device MAC address         |
| 8  | (m) | Attribut:           | Local device R-port1 information |
| 9  | (m) | Attribut:           | Local device R-port2 information |
| 10 | (m) | Attribut:           | Local device protocol version    |
| 11 | (m) | Attribut:           | Local device type                |
| 12 | (m) | Attribut:           | Local device description         |
| 13 | (m) | Attribut:           | Hop count                        |
| 14 | (m) | Attribut:           | FamilyRes frame waiting time     |

|       |     |             |                                              |
|-------|-----|-------------|----------------------------------------------|
| 15    | (m) | Attribut:   | AdvThis frame waiting time                   |
| 16    | (m) | Attribut:   | AckRNMS frame waiting time                   |
| 17    | (m) | Attribut:   | Ring state change timeout                    |
| 18    | (m) | Attribut:   | Diagnostic information (Network, Path Table) |
| 19    | (c) | Contrainte: | Diagnostic information = Network             |
| 19.1  | (m) | Attribut:   | Network topology                             |
| 19.2  | (m) | Attribut:   | Collision count                              |
| 19.3  | (m) | Attribut:   | Device count                                 |
| 19.4  | (m) | Attribut:   | Topology change count                        |
| 19.5  | (m) | Attribut:   | Last topology change time                    |
| 19.6  | (m) | Attribut:   | RNMP device UID                              |
| 19.7  | (m) | Attribut:   | RNMS device UID                              |
| 19.8  | (m) | Attribut:   | LNMP device UID for R-port1                  |
| 19.9  | (m) | Attribut:   | LNMP device UID for R-port2                  |
| 19.10 | (m) | Attribut:   | Network flags                                |
| 20    | (c) | Contrainte: | Diagnostic information = Path Table          |
| 20.1  | (m) | Attribut:   | Peer device address                          |
| 20.2  | (m) | Attribut:   | Peer device hop count for R-port1            |
| 20.3  | (m) | Attribut:   | Peer device hop count for R-port2            |
| 20.4  | (m) | Attribut:   | Preferred R-port for peer device             |
| 20.5  | (m) | Attribut:   | Destination R-port for peer device           |
| 20.6  | (m) | Attribut:   | Peer device state                            |
| 20.7  | (m) | Attribut:   | Peer device MAC address                      |
| 20.8  | (m) | Attribut:   | Peer device R-port1 information              |
| 20.9  | (m) | Attribut:   | Peer device R-port2 information              |
| 20.10 | (m) | Attribut:   | Peer device protocol version                 |
| 20.11 | (m) | Attribut:   | Peer device type                             |
| 20.12 | (m) | Attribut:   | Peer device description                      |
| 20.13 | (m) | Attribut:   | Peer device UID                              |
| 20.14 | (m) | Attribut:   | Device UID for R-port1 of peer device        |
| 20.15 | (m) | Attribut:   | Device UID for R-port2 of peer device        |
| 20.16 | (m) | Attribut:   | In net count of peer device                  |
| 20.17 | (m) | Attribut:   | In net time of peer device                   |
| 20.18 | (m) | Attribut:   | Out net count of peer device                 |
| 20.19 | (m) | Attribut:   | Out net time of peer device                  |

## SERVICES:

|   |     |             |                            |
|---|-----|-------------|----------------------------|
| 1 | (m) | OpsService: | Set Device Information     |
| 2 | (m) | OpsService: | Get Device Information     |
| 3 | (m) | OpsService: | Get Network Information    |
| 4 | (m) | OpsService: | Get Path Table Information |

## 6.3 Attributs

NOTE 1 Le type de données de chaque attribut respecte la définition de la norme CEI 61158-4-21:2010, Article 4.

### Local device address

Cet attribut principal définit l'adresse de dispositif local RRP qui désigne un dispositif unique sur une liaison locale spécifique; sa valeur est contrainte à la plage 0-255.

Type de données: Unsigned16

NOTE 2 Il est admis que l'adresse du dispositif local soit fournie par des paramètres matériels (p. ex., interrupteur rotatif) ou définie de façon logicielle.

#### **Local device flags**

Cet attribut spécifie les indicateurs d'événements survenus dans un dispositif local. La signification des bits est la suivante:

- Bit 0: collision d'adresses de dispositifs
- Bit 1: état modifié du dispositif

Type de données: Unsigned16

#### **Local device state**

Cet attribut spécifie l'état du dispositif local. L'état du dispositif doit avoir l'une des valeurs suivantes:

- 0 = non valide
- 1 = SA (état autonome)
- 2 = LNM (état gestionnaire de réseau linéaire)
- 3 = GD (état dispositif général)
- 4 = RNMP (état gestionnaire de réseau en anneau principal)
- 5 = RNMS (état gestionnaire de réseau en anneau secondaire)

Type de données: Unsigned8

#### **Local device MAC address**

Cet attribut définit l'adresse MAC Ethernet ISO/CEI 8802-3:2000 à 6 octets du dispositif local. Etant donné qu'un dispositif RRP a deux ports MAC Ethernet, il convient que les deux adresses MAC soient identiques.

Type de données: Unsigned48

#### **Local device unique ID**

Cet attribut définit l'identification unique à 8 octets qui identifie un dispositif RRP dans un réseau. Il s'agit d'une combinaison de l'adresse MAC ISO/CEI 8802-3:2000 à 6 octets et de l'adresse du dispositif à 2 octets. Les bits individuels doivent avoir la signification suivante:

- Bit 0 – 15: adresse de dispositif
- Bit 16 – 63: adresse MAC ISO/CEI 8802-3:2000

Type de données: UniqueDeviceID64

#### **Device UID for R-port1**

Cet attribut définit l'UID du dispositif lié via le R-port1. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

#### **Device UID for R-port2**

Cet attribut définit l'UID du dispositif lié via le R-port2. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

#### **Local device R-port1 information**

Cet attribut définit les informations de port pour le R-port1. La signification des bits est la suivante:

- Bit 0: Interruption de la liaison R-port

- Bit 1: Family\_Frame.Cnf reçu depuis le R-port
- Bit 2: en attente de AdvThis\_Frame.Cnf depuis le R-port
- Bit 3: en attente de MediaLinked\_Frame.Ind depuis le R-port
- Bit 4: confirmation de l'état du dispositif

Type de données: Unsigned8

#### **Local device R-port2 information**

Cet attribut définit les informations de port pour le R-port2. Il doit avoir l'une des valeurs définies pour les informations du R-port1 relatives au dispositif local.

Type de données: Unsigned8

#### **Local device protocol version**

Cet attribut définit la version du protocole RRP du dispositif local. Les bits individuels doivent avoir la signification suivante:

- Bit 0 – 1: version majeure
- Bit 2 – 4: version mineure
- Bit 5 – 7: réservé

Type de données: Unsigned8

#### **Local device type**

Cet attribut définit le type de dispositif local, qui représente la fonction générale du dispositif. Les bits individuels doivent avoir la signification suivante:

- Bit 0 – 7: type de dispositif général
- Bit 8 – 15: type de dispositif spécifique à l'application

Type de données: Unsigned16

#### **Local device description**

Cet attribut définit une description du dispositif local et contient toutes les chaînes définies par l'utilisateur à l'aide du service Set Device Information.

Type de données: VisibleString[16]

#### **Hop count**

Cet attribut définit le nombre de dispositifs compris entre deux dispositifs. Lorsque le dispositif RRP reçoit la trame NCM, il enregistre la valeur du nombre de sauts reçue, puis incrémente le nombre de sauts dans la trame reçue par paliers de 1 et transmet la trame par l'intermédiaire de l'autre R-port. De cette manière, chaque dispositif établit sa propre table de chemins avec un nombre de sauts spécifique au R-port1 et un nombre de sauts spécifique au R-port2.

Type de données: Unsigned16

#### **FamilyRes frame waiting time**

Cet attribut définit l'intervalle de temps entre l'envoi de la trame FamilyReq et la réception de la trame FamilyRes. Cet attribut doit être configuré par l'utilisateur à l'aide du service Set Device Information.

Type de données: Unsigned32

#### **AdvThis frame waiting time**

Cet attribut définit l'intervalle de temps entre l'envoi de la trame MediaLinked et la réception de la trame AdvThis. Cet attribut doit être configuré par l'utilisateur à l'aide du service Set Device Information.

Type de données: Unsigned32

**AckRNMS frame waiting time**

Cet attribut définit l'intervalle de temps entre l'envoi de la trame RingStart et la réception de la trame AckRNMS. Cet attribut doit être configuré par l'utilisateur à l'aide du service Set Device Information.

Type de données: Unsigned32

**Ring state change timeout**

Cet attribut définit le délai d'attente nécessaire pour générer l'événement permettant de modifier l'état du dispositif RNMP. Cet attribut doit être configuré par l'utilisateur à l'aide du service Set Device Information.

Type de données: Unsigned32

**Diagnostic information**

Cet attribut définit le type d'informations de diagnostic. Ce type doit avoir les deux valeurs du réseau et de la table de chemins.

Type de données: Unsigned8

**Network topology**

Cet attribut définit le type de la topologie de réseau. Il doit avoir l'une des valeurs suivantes:

- 0 = non valide
- 1 = NET\_TPG\_SA (autonome)
- 2 = NET\_TPG\_LINE (topologie linéaire)
- 3 = NET\_TPG\_RING (topologie en anneau)

Type de données: Unsigned8

**Collision count**

Cet attribut définit le nombre de collisions d'adresses pour les dispositifs à distance. Les valeurs sont comprises dans la plage 0-255.

Type de données: Unsigned8

NOTE 3 La valeur est incrémentée par la machine de protocole RRP lorsqu'une collision d'adresses de dispositifs à distance est détectée. Cette valeur est décrétementée lorsque la collision est supprimée.

**Device count**

Cet attribut définit le nombre total de dispositifs sur le réseau. Les valeurs sont comprises dans la plage 1-255.

Type de données: Unsigned16

**Topology change count**

Cet attribut définit le nombre de modifications de topologie. Les valeurs sont comprises dans la plage 0-65 535.

Type de données: Unsigned16

NOTE 4 La valeur est incrémentée par la machine de protocole RRP lorsque le réseau passe d'une topologie en anneau à une topologie linéaire, ou inversement.

**Last topology change time**

Cet attribut définit la date et l'heure de la dernière modification de topologie du réseau.

Type de données: TIMEOFDAY

**RNMP device UID**

Cet attribut définit l'UID de dispositif sélectionné en tant que RNMP sur le réseau. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

### **RNMS device UID**

Cet attribut définit l'UID du dispositif sélectionné en tant que RNMS sur le réseau. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

### **LNM device UID for R-port1**

Cet attribut définit l'UID du dispositif sélectionné en tant que LNM dans le sens R-port1. Dans un réseau linéaire RRP, les deux dispositifs d'extrémité sont automatiquement sélectionnés en tant que LNM. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

### **LNM device UID for R-port2**

Cet attribut définit l'UID du dispositif sélectionné en tant que LNM dans le sens R-port2. Dans un réseau linéaire RRP, les deux dispositifs d'extrémité sont automatiquement sélectionnés en tant que LNM. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

### **Network flags**

Cet attribut définit les indicateurs d'événements survenus sur le réseau. La signification des bits est la suivante:

- Bit 0: topologie de réseau modifiée
- Bit 1: détection d'une collision d'adresses de dispositifs sur le réseau
- Bit 2: intégration d'un nouveau dispositif au réseau
- Bit 3: retrait d'un dispositif du réseau

Type de données: Unsigned16

NOTE 5 Chaque bit est défini par la machine de protocole RRP lorsque l'événement qui lui correspond se produit.

### **Peer device address**

Cet attribut définit l'adresse du dispositif homologue, stockée dans la table de chemins. La valeur est contrainte à la plage 0-255.

Type de données: Unsigned16

### **Peer device hop count for R-port1**

Cet attribut définit le nombre de transferts de trames nécessaires pour l'envoi d'une trame depuis le dispositif local vers le dispositif homologue via le R-port1.

Type de données: Unsigned16

### **Peer device hop count for R-port2**

Cet attribut définit le nombre de transferts de trames nécessaires pour l'envoi d'une trame depuis le dispositif local vers le dispositif homologue via le R-port2.

Type de données: Unsigned16

### **Preferred R-port for peer device**

Cet attribut définit le R-port préférentiel pour l'envoi d'une trame depuis le dispositif local vers le dispositif homologue sans prise en compte du RNMP ou du RNMS. Il doit avoir l'une des valeurs suivantes:

- 0 = non valide
- 1 = R-port1
- 2 = R-port2

Type de données: Unsigned8

NOTE 6 Il est défini comme le R-port ayant la valeur de nombre de sauts la plus faible pour le dispositif homologue. Si les nombres de sauts du R-port1 et du R-port2 sont identiques, le R-port1 est sélectionné en tant que Rport préférentiel.

#### **Destination R-port for peer device**

Cet attribut définit le R-port de destination pour l'envoi d'une trame depuis le dispositif local vers le dispositif homologue.

Type de données: Unsigned8

NOTE 7 Sur un réseau linéaire, cette variable a la même valeur que le R-port préférentiel. Cependant, sur un réseau en anneau, cette variable est déterminée en fonction des positions du RNMP et du RNMS, car il est admis que le chemin préférentiel soit bloqué par le RNMP ou le RNMS. Dans ce cas, le R-port de destination est sélectionné comme étant l'autre R-port. Il convient qu'il ait l'une des valeurs définies pour le R-port préférentiel.

#### **Peer device state**

Cet attribut définit l'état du dispositif homologue, stocké dans la table de chemins. Il doit avoir l'une des valeurs définies pour l'état du dispositif local.

Type de données: Unsigned8

#### **Peer device MAC address**

Cet attribut définit l'adresse MAC Ethernet ISO/CEI 8802-3:2000 à 6 octets du dispositif homologue, stockée dans la table de chemins.

Type de données: Unsigned48

#### **Peer device R-port1 information**

Cet attribut définit les informations du R-port1 applicables au dispositif homologue, stockées dans la table de chemins. Il doit avoir l'une des valeurs définies pour les informations de R-port1 applicables au dispositif local.

Type de données: Unsigned8

#### **Peer device R-port2 information**

Cet attribut définit les informations de R-port2 applicables au dispositif homologue, stockées dans la table de chemins. Il doit avoir l'une des valeurs définies pour les informations de R-port1 applicables au dispositif local.

Type de données: Unsigned8

#### **Peer device protocol version**

Cet attribut définit la version de protocole RRP applicable au dispositif homologue, stockée dans la table de chemins. Les bits individuels doivent avoir la même signification que la version de protocole du dispositif local.

Type de données: Unsigned8

#### **Peer device type**

Cet attribut définit le type du dispositif homologue spécifique à une application, stocké dans la table de chemins. Les bits individuels doivent avoir la même signification que le type de dispositif local.

Type de données: Unsigned16

#### **Peer device description**

Cet attribut définit la description du dispositif homologue, stockée dans la table de chemins.

Type de données: VisibleString[16]

#### **Peer device UID**

Cet attribut définit l'identification unique du dispositif homologue, stockée dans la table de chemins. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

#### **Device UID for R-port1 of peer device**

Cet attribut définit l'UID du dispositif lié par l'intermédiaire du R-port1 du dispositif homologue, stockée dans la table de chemins. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

#### **Device UID for R-port2 of peer device**

Cet attribut définit l'UID du dispositif lié par l'intermédiaire du R-port2 du dispositif homologue, stockée dans la table de chemins. Les bits individuels doivent avoir la même signification que l'identification unique de dispositif local.

Type de données: UniqueDeviceID64

#### **In net count of peer device**

Cet attribut définit le nombre de fois où le dispositif homologue a intégré le réseau.

Type de données: Unsigned16

NOTE 8 Lorsqu'un réseau linéaire est fusionné à un réseau existant, les variables des dispositifs nouvellement intégrés sont incrémentées en même temps.

#### **In net time of peer device**

Cet attribut définit la date et l'heure de la dernière intégration du dispositif homologue au réseau.

Type de données: TIMEOFDAY

#### **Out net count of peer device**

Cet attribut définit le nombre de fois où le dispositif homologue a été déconnecté du réseau. Lorsqu'un dispositif ou un groupe de dispositifs est déconnecté du réseau, les variables des dispositifs ainsi déconnectés sont incrémentées en même temps.

Type de données: Unsigned16

#### **Out net time of peer device**

Cet attribut définit la date et l'heure de la dernière déconnexion du dispositif du réseau.

Type de données: TIMEOFDAY

## **7 Spécifications des services RRP**

### **7.1 Set device information**

Ce service permet d'affecter de nouvelles valeurs aux variables des informations relatives au dispositif local.

Les paramètres de ce service sont spécifiés dans le Tableau 8.

**Tableau 8 – Paramètres du service set device information**

| Nom de paramètre              | Req | Ind  | Rsp | Cnf  |
|-------------------------------|-----|------|-----|------|
| Argument                      | M   | M(=) |     |      |
| Service ID                    | M   | M(=) |     |      |
| Invoke ID                     | M   | M(=) |     |      |
| Local device address          | M   | M(=) |     |      |
| Local device MAC address      | M   | M(=) |     |      |
| Local device protocol version | M   | M(=) |     |      |
| Local device type             | M   | M(=) |     |      |
| Local device description      | M   | M(=) |     |      |
| FamilyRes frame waiting time  | M   | M(=) |     |      |
| AdvThis frame waiting time    | M   | M(=) |     |      |
| AckRNMS frame waiting time    | M   | M(=) |     |      |
| Ring state change timeout     | M   | M(=) |     |      |
| Result(+)                     |     |      | S   | S(=) |
| Service ID                    |     |      | M   | M(=) |
| Invoke ID                     |     |      | M   | M(=) |
| Status Code                   |     |      | M   | M(=) |
| Result(-)                     |     |      | S   | S(=) |
| Service ID                    |     |      | M   | M(=) |
| Invoke ID                     |     |      | M   | M(=) |
| Status Code                   |     |      | M   | M(=) |

**Argument**

L'argument transmet les paramètres spécifiques au service applicables à la demande de service.

**Service ID**

Ce paramètre contient des informations suffisantes pour identifier au niveau local le dispositif RRP à utiliser pour transmettre le service.

**Invoke ID**

Ce paramètre identifie l'invocation du service.

**Local device address**

Ce paramètre contient la valeur de l'adresse du dispositif RRP.

**Local device MAC address**

Ce paramètre contient l'adresse MAC du dispositif RRP.

**Local device protocol version**

Ce paramètre contient la version de protocole du dispositif RRP.

**Local device type**

Ce paramètre contient le type de dispositif RRP, qui représente la fonction générale du dispositif RRP.

**Local device description**

Ce paramètre contient une description du dispositif RRP.

**FamilyRes frame waiting time**

Ce paramètre contient la valeur du temps d'attente de la trame FamilyRes applicable au dispositif RRP.

**AdvThis frame waiting time**

Ce paramètre contient la valeur du temps d'attente de la trame AdvThis applicable au dispositif RRP.

**AckRNMS frame waiting time**

Ce paramètre contient la valeur du temps d'attente de la trame AckRNMS applicable au dispositif RRP.

**Ring state change timeout**

Ce paramètre contient la valeur du délai d'attente nécessaire pour passer à un état de topologie en anneau applicable au dispositif RRP.

**Result(+)**

Ce paramètre indique que la demande de service a réussi.

**Service ID**

Ce paramètre contient des informations suffisantes pour identifier au niveau local le dispositif RRP à utiliser pour transmettre le service.

**Invoke ID**

Ce paramètre identifie l'invocation du service.

**Status Code**

Ce paramètre indique si le traitement du service a réussi. Si une erreur se produit, il en indique le type.

Type de données: Unsigned8

**Result(-)**

Ce paramètre indique que la demande de service a échoué.

**Service ID**

Ce paramètre contient des informations suffisantes pour identifier au niveau local le dispositif RRP à utiliser pour transmettre le service.

**Invoke ID**

Ce paramètre identifie l'invocation du service.

**Status Code**

Ce paramètre indique si le traitement du service a réussi. Si une erreur se produit, il en indique le type.

Type de données: Unsigned8

**7.2 Get device information**

Ce service permet d'obtenir des informations sur le dispositif local à partir du dispositif RRP.

Les paramètres de ce service sont spécifiés dans le Tableau 9.

**Tableau 9 – Paramètres du service get device information**

| Nom de paramètre                 | Req | Ind  | Rsp | Cnf  |
|----------------------------------|-----|------|-----|------|
| Argument                         | M   | M(=) |     |      |
| Service ID                       | M   | M(=) |     |      |
| Invoke ID                        | M   | M(=) |     |      |
| Result(+)                        |     |      | S   | S(=) |
| Service ID                       |     |      | M   | M(=) |
| Invoke ID                        |     |      | M   | M(=) |
| Local device address             |     |      | M   | M(=) |
| Local device flags               |     |      | M   | M(=) |
| Local device state               |     |      | M   | M(=) |
| Local device unique ID           |     |      | M   | M(=) |
| Device UID for R-port1           |     |      | M   | M(=) |
| Device UID for R-port2           |     |      | M   | M(=) |
| Local device MAC address         |     |      | M   | M(=) |
| Local device R-port1 information |     |      | M   | M(=) |
| Local device R-port2 information |     |      | M   | M(=) |
| Local device protocol version    |     |      | M   | M(=) |
| Local device type                |     |      | M   | M(=) |
| Local device description         |     |      | M   | M(=) |
| FamilyRes frame waiting time     |     |      | M   | M(=) |
| AdvThis frame waiting time       |     |      | M   | M(=) |
| AckRNMS frame waiting time       |     |      | M   | M(=) |
| Ring state change timeout        |     |      | M   | M(=) |
| Result(-)                        |     |      | S   | S(=) |
| Service ID                       |     |      | M   | M(=) |
| Invoke ID                        |     |      | M   | M(=) |
| Status Code                      |     |      | M   | M(=) |

**Argument**

L'argument transmet les paramètres spécifiques au service applicables à la demande de service.

**Service ID**

Ce paramètre est défini en 7.1.

**Invoke ID**

Ce paramètre est défini en 7.1.

**Result(+)**

Ce paramètre indique que la demande de service a réussi.

**Service ID**

Ce paramètre est défini en 7.1.

**Invoke ID**

Ce paramètre est défini en 7.1.

**Local device address**

Ce paramètre est défini en 7.1.

**Local device flags**

Ce paramètre contient les indicateurs du dispositif RRP.

**Local device state**

Ce paramètre contient l'état du dispositif RRP.

**Local device unique ID**

Ce paramètre contient l'identification unique du dispositif RRP.

**Device UID for R-port1**

Ce paramètre contient l'UID du dispositif lié via le R-port1.

**Device UID for R-port2**

Ce paramètre contient l'UID du dispositif lié via le R-port2.

**Local device MAC address**

Ce paramètre est défini en 7.1.

**Local device R-port1 information**

Ce paramètre contient les informations sur le R-port1 applicables au dispositif RRP.

**Local device R-port2 information**

Ce paramètre contient les informations sur le R-port2 applicables au dispositif RRP.

**Local device protocol version**

Ce paramètre contient la version de protocole du dispositif RRP.

**Local device type**

Ce paramètre est défini en 7.1.

**Local device description**

Ce paramètre est défini en 7.1.

**FamilyRes frame waiting time**

Ce paramètre est défini en 7.1.

**AdvThis frame waiting time**

Ce paramètre est défini en 7.1.

**AckRNMS frame waiting time**

Ce paramètre est défini en 7.1.

**Ring state change timeout**

Ce paramètre est défini en 7.1.

**Result(-)**

Ce paramètre indique que la demande de service a échoué.

**Service ID**

Ce paramètre est défini en 7.1.

**Invoke ID**

Ce paramètre est défini en 7.1.

**Status Code**

Ce paramètre est défini en 7.1.

**7.3 Get network information**

Ce service permet d'obtenir des informations sur le réseau à partir du dispositif RRP.

Les paramètres de ce service sont spécifiés dans le Tableau 10.

**Tableau 10 – Paramètres du service get network information**

| Nom de paramètre           | Req | Ind  | Rsp | Cnf  |
|----------------------------|-----|------|-----|------|
| Argument                   | M   | M(=) |     |      |
| Service ID                 | M   | M(=) |     |      |
| Invoke ID                  | M   | M(=) |     |      |
| Result(+)                  |     |      | S   | S(=) |
| Service ID                 |     |      | M   | M(=) |
| Invoke ID                  |     |      | M   | M(=) |
| Network topology           |     |      | M   | M(=) |
| Collision count            |     |      | M   | M(=) |
| Device count               |     |      | M   | M(=) |
| Topology change count      |     |      | M   | M(=) |
| Last topology change time  |     |      | M   | M(=) |
| RNMP device UID            |     |      | M   | M(=) |
| RNMS device UID            |     |      | M   | M(=) |
| LNM device UID for R-port1 |     |      | M   | M(=) |
| LNM device UID for R-port2 |     |      | M   | M(=) |
| Network flags              |     |      | M   | M(=) |
| Result(-)                  |     |      | S   | S(=) |
| Service ID                 |     |      | M   | M(=) |
| Invoke ID                  |     |      | M   | M(=) |
| Status Code                |     |      | M   | M(=) |

### Argument

L'argument transmet les paramètres spécifiques au service applicables à la demande de service.

#### Service ID

Ce paramètre est défini en 7.1.

#### Invoke ID

Ce paramètre est défini en 7.1.

### Result(+)

Ce paramètre indique que la demande de service a réussi.

#### Service ID

Ce paramètre est défini en 7.1.

#### Invoke ID

Ce paramètre est défini en 7.1.

#### Network topology

Ce paramètre contient le type de la topologie de réseau.

#### Collision count

Ce paramètre contient le nombre de collisions d'adresses pour les dispositifs à distance.

#### **Device count**

Ce paramètre contient le nombre total de dispositifs sur le réseau.

#### **Topology change count**

Ce paramètre contient le nombre de modifications de topologie.

#### **Last topology change time**

Ce paramètre contient la date et l'heure de la dernière modification de topologie de réseau.

#### **RNMP device UID**

Ce paramètre contient l'UID du dispositif sélectionné en tant que RNMP sur le réseau.

#### **RNMS device UID**

Ce paramètre contient l'UID du dispositif sélectionné en tant que RNMS sur le réseau.

#### **LNM device UID for R-port1**

Ce paramètre contient l'UID du dispositif sélectionné en tant que LNM dans le sens R-port1.

#### **LNM device UID for R-port2**

Ce paramètre contient l'UID du dispositif sélectionné en tant que LNM dans le sens R-port2.

#### **Network flags**

Ce paramètre contient les indicateurs d'événements survenus sur le réseau.

#### **Result(-)**

Ce paramètre indique que la demande de service a échoué.

#### **Service ID**

Ce paramètre est défini en 7.1.

#### **Invoke ID**

Ce paramètre est défini en 7.1.

#### **Status Code**

Ce paramètre est défini en 7.1.

### **7.4 Get path table information**

Ce service permet d'obtenir des informations sur la table de chemins à partir du dispositif RRP. La table de chemins est gérée par la machine de protocole RRP. Il se présente sous la forme d'un tableau d'ensemble complété à l'aide d'informations relatives au chemin de chaque dispositif RRP du réseau. La taille maximale de la table de chemins correspond à une fonction MAX\_ADDR, comme suit:

**Path table: Array[n] of device's path information, n = MAX\_ADDR + 1**

NOTE MAX\_ADDR correspond à l'adresse maximale de dispositif. Elle est définie par la machine de protocole RRP. La plage applicable à cette variable est 1-255. La valeur par défaut de cette variable est 255. Cette variable indique également le nombre maximal d'entrées de la table de chemins dans la NMIB. Les valeurs comprises dans la plage 256-65 535 sont réservées.

Les paramètres de ce service sont spécifiés dans le Tableau 11.

**Tableau 11 – Paramètres du service get path table Information**

| Nom de paramètre                      | Req | Ind  | Rsp | Cnf  |
|---------------------------------------|-----|------|-----|------|
| Argument                              | M   | M(=) |     |      |
| Service ID                            | M   | M(=) |     |      |
| Invoke ID                             | M   | M(=) |     |      |
| Result(+)                             |     |      | S   | S(=) |
| Service ID                            |     |      | M   | M(=) |
| Invoke ID                             |     |      | M   | M(=) |
| Peer device address                   |     |      | M   | M(=) |
| Peer device hop count for R-port1     |     |      | M   | M(=) |
| Peer device hop count for R-port2     |     |      | M   | M(=) |
| Preferred R-port for peer device      |     |      | M   | M(=) |
| Destination R-port for peer device    |     |      | M   | M(=) |
| Peer device state                     |     |      | M   | M(=) |
| Peer device MAC address               |     |      | M   | M(=) |
| Peer device R-port1 information       |     |      | M   | M(=) |
| Peer device R-port2 information       |     |      | M   | M(=) |
| Peer device protocol version          |     |      | M   | M(=) |
| Peer device type                      |     |      | M   | M(=) |
| Peer device description               |     |      | M   | M(=) |
| Peer device UID                       |     |      | M   | M(=) |
| Device UID for R-port1 of peer device |     |      | M   | M(=) |
| Device UID for R-port2 of peer device |     |      | M   | M(=) |
| In net count of peer device           |     |      | M   | M(=) |
| In net time of peer device            |     |      | M   | M(=) |
| Out net count of peer device          |     |      | M   | M(=) |
| Out net time of peer device           |     |      | M   | M(=) |
| Result(-)                             |     |      | S   | S(=) |
| Service ID                            |     |      | M   | M(=) |
| Invoke ID                             |     |      | M   | M(=) |
| Status Code                           |     |      | M   | M(=) |

**Argument**

L'argument transmet les paramètres spécifiques au service applicables à la demande de service.

**Service ID**

Ce paramètre est défini en 7.1.

**Invoke ID**

Ce paramètre est défini en 7.1.

**Result(+)**

Ce paramètre indique que la demande de service a réussi.

**Service ID**

Ce paramètre est défini en 7.1.

**Invoke ID**

Ce paramètre est défini en 7.1.

**Peer device address**

Ce paramètre contient la valeur de l'adresse du dispositif homologue RRP.

**Peer device hop count for R-port1**

Ce paramètre contient le nombre de transferts de trames nécessaires pour l'envoi d'une trame depuis le dispositif local vers le dispositif homologue via le R-port1.

**Peer device hop count for R-port2**

Ce paramètre contient le nombre de transferts de trames nécessaires pour l'envoi d'une trame depuis le dispositif local vers le dispositif homologue via le R-port2.

**Preferred R-port for peer device**

Ce paramètre contient le R-port préférentiel pour l'envoi d'une trame depuis le dispositif local vers le dispositif homologue sans prise en compte du RNMP ou du RNMS.

**Destination R-port for peer device**

Ce paramètre contient le R-port de destination pour l'envoi d'une trame depuis le dispositif local vers le dispositif homologue.

**Peer device state**

Ce paramètre contient l'état du dispositif homologue, stocké dans la table de chemins.

**Peer device MAC address**

Ce paramètre contient l'adresse MAC Ethernet ISO/CEI 8802-3:2000 à 6 octets du dispositif homologue, stockée dans la table de chemins.

**Peer device R-port1 information**

Ce paramètre contient les informations du R-port1 applicables au dispositif homologue, stockées dans la table de chemins.

**Peer device R-port2 information**

Ce paramètre contient les informations du R-port2 applicables au dispositif homologue, stockées dans la table de chemins.

**Peer device protocol version**

Ce paramètre contient la version de protocole RRP applicable au dispositif homologue, stockée dans la table de chemins.

**Peer device type**

Ce paramètre contient le type du dispositif homologue spécifique à une application, stocké dans la table de chemins.

**Peer device description**

Ce paramètre contient la description du dispositif homologue, stockée dans la table de chemins.

**Peer device UID**

Ce paramètre contient l'identification unique du dispositif homologue, stockée dans la table de chemins.

**Device UID for R-port1 of peer device**

Ce paramètre contient l'UID du dispositif lié par l'intermédiaire du R-port1 du dispositif homologue, stockée dans la table de chemins.

**Device UID for R-port2 of peer device**

Ce paramètre contient l'UID du dispositif lié par l'intermédiaire du R-port2 du dispositif homologue, stockée dans la table de chemins.

**In net count of peer device**

Ce paramètre contient le nombre de fois où le dispositif homologue a intégré le réseau.

**In net time of peer device**

Ce paramètre contient la date et l'heure de la dernière intégration du dispositif homologue au réseau.

**Out net count of peer device**

Ce paramètre contient le nombre de fois où le dispositif homologue a été déconnecté du réseau.

**Out net time of peer device**

Ce paramètre contient la date et l'heure de la dernière déconnexion du dispositif du réseau.

**Result(-)**

Ce paramètre indique que la demande de service a échoué.

**Service ID**

Ce paramètre est défini en 7.1.

**Invoke ID**

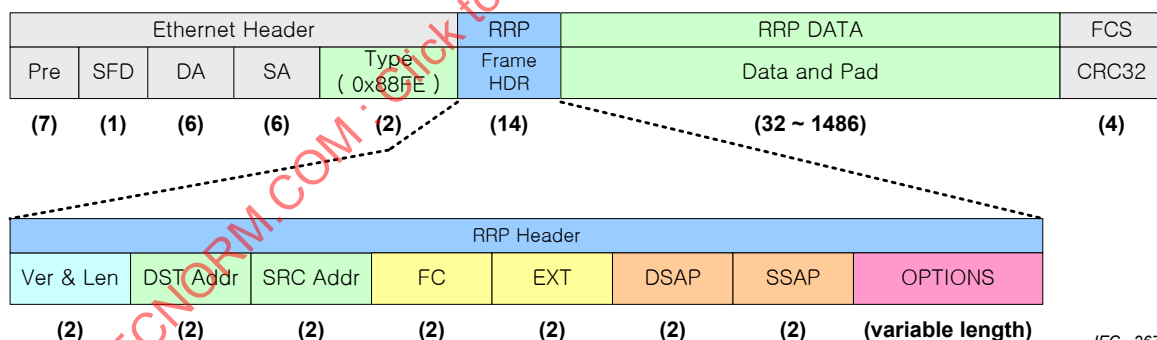
Ce paramètre est défini en 7.1.

**Status Code**

Ce paramètre est défini en 7.1.

**8 Spécification de protocole RRP****8.1 Généralités**

Le codage et le décodage des champs dans l'unité de données de protocole de liaison de données (DLPDU) sont encapsulés dans le champ de données d'une trame MAC, comme indiqué par l'ISO/CEI 8802-3:2000, Article 3. La valeur du champ Length/Type est 0x88FE, laquelle est autorisée et enregistrée en tant que numéro d'identification de protocole par l'Autorité d'enregistrement d'IEEE pour identifier une trame RRP. La Figure 18 présente la structure de DLPDU RRP.



IEC 2678/11

**Légende**

| Anglais         | Français         |
|-----------------|------------------|
| Ethernet Header | En-tête Ethernet |
| Pre             | Pre              |
| SFD             | SFD              |
| DA              | DA               |
| SA              | SA               |
| Type            | Type             |
| RRP             | RRP              |
| RRP DATA        | DONNEES RRP      |
| Data and Pad    | Data and Pad     |
| FCS             | FCS              |

| Anglais           | Français         |
|-------------------|------------------|
| RRP Header        | En-tête RRP      |
| Ver & Len         | Ver & Len        |
| DST Addr          | DST Addr         |
| SRC Addr          | SRC Addr         |
| FC                | FC               |
| EXT               | EXT              |
| DSAP              | DSAP             |
| SSAP              | SSAP             |
| OPTIONS           | OPTIONS          |
| (variable length) | (long. variable) |

**Figure 18 – Format de trame MAC courante pour DLPDU RRP**

## 8.2 En-tête Ethernet

### 8.2.1 Preamble

Ce champ doit être codé conformément à l'ISO/CEI 8802-3:2000, Article 3.

### 8.2.2 Start frame delimiter

Ce champ doit être codé conformément à l'ISO/CEI 8802-3:2000, Article 3.

### 8.2.3 Destination MAC address

Ce champ doit être codé conformément à l'ISO/CEI 8802-3:2000, Article 3. Il spécifie le ou les dispositifs auxquels la trame est destinée. Il est admis qu'il s'agisse d'une adresse individuelle ou multidiffusion (y compris de diffusion).

### 8.2.4 Source MAC address

Ce champ doit être codé conformément à l'ISO/CEI 8802-3:2000, Article 3.

### 8.2.5 Length/Type

Ce champ doit être codé conformément à l'ISO/CEI 8802-3:2000, Article 3 "Media access control frame structure". Pour être identifiée en tant que trame RRP, la valeur du champ Length/Type est définie sur 0x88FE, laquelle est autorisée et enregistrée en tant que numéro d'identification de protocole pour RRP par l'Autorité d'enregistrement d'IEEE. Toutes les trames avec une valeur différente de 0x88FE sont identiques à la trame dans ISO/CEI 8802-3:2000, Article 3, et sont traitées comme des trames de données sporadiques de réseaux de terrain RRP. Pour RRP, la valeur doit être définie conformément au Tableau 12.

**Tableau 12 – Champ Length/Type RRP**

| Valeur (hexadécimale) | Signification |
|-----------------------|---------------|
| 0x88FE                | RRP-PDU       |

### 8.3 Codage de RRP\_FrameHDR

#### 8.3.1 Version and length

Ce champ indique la version de protocole et la longueur applicables au protocole RRP. Il doit être codé en tant que type de données Unsigned16 et les bits individuels doivent avoir la signification suivante:

##### Bit 0 – 10: length

La valeur indique le nombre d'octets de la trame intégrant le champ FCS.

##### Bit 11 – 15: version

La version est représentée par deux bits pour la version majeure et par trois bits pour la version mineure. Ce champ doit être codé avec les valeurs conformément au Tableau 13.

**Tableau 13 – Version**

| Nom de champ | Position    | Signification                    |
|--------------|-------------|----------------------------------|
| Major        | Bit 14 – 15 | Version majeure du protocole RRP |
| Minor        | Bit 11 – 13 | Version mineure du protocole RRP |

#### 8.3.2 DST\_addr

##### 8.3.2.1 Généralités

Ce champ indique l'identificateur du dispositif RRP de destination du nœud vers lequel la trame est envoyée. Il doit être codé en tant que type de données Unsigned16 et défini conformément au Tableau 14.

**Tableau 14 – DST\_addr**

| Valeur (hexadécimale) | Signification                                    |
|-----------------------|--------------------------------------------------|
| 0xFFFF                | adresse de diffusion                             |
| 0xFFFE                | adresse de contrôle réseau (C_NCM_ADDR)          |
| 0xFFFD – 0xFFDE       | adresse multidiffusion définie par l'utilisateur |
| 0xFFDD                | adresse non valide                               |
| 0x0100 – 0xFFDC       | réservé                                          |
| 0x0000 – 0x00FF       | adresse normale de dispositif RRP                |

##### 8.3.2.2 Adresse de diffusion

Si l'identificateur du dispositif RRP de destination est 0xFFFF, le champ Destination MAC Address contient l'adresse MAC de diffusion ISO/CEI 8802-3:2000.

##### 8.3.2.3 Adresse de contrôle réseau

Le protocole RRP définit une adresse MAC spécifique, 00-E0-91-02-05-99 (NCM\_MAC\_ADDR), en vue de partager des informations de gestion de réseau. Tous les messages reçus via NCM\_MAC\_ADDR mettent à jour les informations de gestion de réseau.

Si l'adresse du dispositif RRP de destination est 0xFFFE (NCM\_ADDR), le champ Destination MAC Address contient NCM\_MAC\_ADDR. Cependant, tous les messages NCM, à l'exception de NCM\_ACK\_RNMS et de NCM\_CHECK\_RNMS, sont transmis à l'aide de NCM\_ADDR comme adresse du dispositif RRP de destination.

### 8.3.2.4 Adresse multidiffusion définie par l'utilisateur

Une adresse multidiffusion définie par l'utilisateur permet d'indiquer plusieurs destinataires. Cependant, l'adressage multidiffusion défini par l'utilisateur n'est pas une fonction obligatoire dans la présente Norme. Il est conçu en vue d'une utilisation dans un système d'applications spécifiques nécessitant une communication multidiffusion. Par conséquent, l'adressage multidiffusion défini par l'utilisateur n'est pas interopérable entre des dispositifs hétérogènes. La plage d'adresse du dispositif RRP de destination 0xFFFD-0xFFDE permet de spécifier l'adresse multidiffusion définie par l'utilisateur. Cependant, la méthode d'utilisation applicable à l'adresse multidiffusion définie par l'utilisateur n'est pas spécifiée dans la présente Norme et est considérée comme relevant d'une responsabilité locale. Cette spécification ne restreint pas l'utilisation des adresses multidiffusion définies par l'utilisateur et n'est pas non plus une fonction obligatoire.

### 8.3.3 SRC\_addr

Ce champ indique l'adresse du dispositif RRP source du nœud générant la trame. Il doit être codé en tant que type de données Unsigned16 et défini conformément au Tableau 15.

Tableau 15 – SRC\_addr

| Valeur<br>(hexadécimale) | Signification                    |
|--------------------------|----------------------------------|
| 0x0000 – 0xFFFF          | Adresse de dispositif RRP source |

### 8.3.4 Frame Control (FC)

#### 8.3.4.1 Généralités

Ce champ indique les informations concernant le contrôle de trames. Il doit être codé en tant que type de données Unsigned16. Il s'agit d'un jeu de bits codé comme suit:

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

#### Bit 0 — 7: Network Control Message Type (NCMT)

Ce champ doit être codé avec les valeurs conformément au Tableau 16.

Tableau 16 – Network control message type

| Valeur<br>(hexadécimale) | Signification    | Utilisation |
|--------------------------|------------------|-------------|
| 0x00                     | Réservé          | —           |
| 0x01                     | NCM_FAMILY_REQ   | Obligatoire |
| 0x02                     | NCM_FAMILY_RES   | Obligatoire |
| 0x03                     | NCM_MEDIA_LINKED | Obligatoire |
| 0x04                     | NCM_ADV_THIS     | Obligatoire |
| 0x05                     | NCM_LINE_START   | Obligatoire |
| 0x06                     | NCM_RING_START   | Obligatoire |
| 0x07                     | NCM_ACK_RNMS     | Obligatoire |
| 0x08                     | NCM_CHECK_RNMS   | Obligatoire |
| 0x09-0xFF                | Réservé          | —           |

**Bit 8 — 11: Type of Service (ToS)**

Ce champ doit être codé avec les valeurs conformément au Tableau 17.

**Tableau 17 – Type of service**

| Valeur<br>(hexadécimale) | Signification                    | Utilisation |
|--------------------------|----------------------------------|-------------|
| 0x00                     | Message de contrôle réseau (NCM) | Obligatoire |
| 0x01                     | Demande de service non confirmée | Facultatif  |
| 0x02-0x0F                | Réservé                          | —           |

**Bit 12 — 13: Priority (PRI)**

Ce champ doit être codé avec les valeurs conformément au Tableau 18.

**Tableau 18 – Priority**

| Valeur<br>(hexadécimale) | Signification          |
|--------------------------|------------------------|
| 0x00                     | priorité la plus basse |
| 0x01-0x02                | ...                    |
| 0x03                     | Priorité la plus haute |

**Bit 14: Reserved (RES)**

Ce champ doit être défini sur 0.

**Bit 15: Validation of Extension code (VoE)**

Ce champ doit être codé avec les valeurs conformément au Tableau 19.

**Tableau 19 – Validation of extension code**

| Valeur<br>(décimale) | Signification                |
|----------------------|------------------------------|
| 0                    | Le code EXT n'est pas valide |
| 1                    | Le code EXT est valide       |

**8.3.4.2 Network control message type**

Ce champ indique le type de message de contrôle réseau. Les valeurs individuelles doivent avoir les significations suivantes:

**0x01: NCM\_FAMILY\_REQ**

Cette valeur permet de demander au dispositif récemment connecté via un R-port s'il s'agit d'un dispositif RRP. Ce message est transmis via le R-port qui a généré les événements d'établissement de liaison PHY. Ce message ne doit pas être transféré vers l'autre port.

**0x02: NCM\_FAMILY\_RES**

Cette valeur permet de confirmer si le destinataire est un dispositif RRP, ce, lorsque le destinataire reçoit le message NCM\_FAMILY\_REQ du dispositif récemment lié. Ce message est transmis via le R-port utilisé pour recevoir le message NCM\_FAMILY\_REQ. Ce message ne doit pas être transféré vers l'autre port.

#### **0x03: NCM\_MEDIA\_LINKED**

Cette valeur permet d'indiquer qu'une nouvelle liaison RRP a été établie via le R-port. Ce message est transmis via le R-port récemment activé. Lorsque le destinataire reçoit ce message, il incrémente le nombre de sauts dans la trame et transfère celle-ci via l'autre R-port. Ce message est supprimé par le LNM ou le dispositif ayant généré le message.

#### **0x04: NCM\_ADV\_THIS**

Cette valeur permet de transmettre les informations relatives au dispositif local du destinataire lorsque ce dernier reçoit le message NCM\_MEDIA\_LINKED du dispositif récemment lié. Ce message est transmis via le R-port utilisé pour recevoir le message NCM\_MEDIA\_LINKED.

#### **0x05: NCM\_LINE\_START**

Cette valeur permet de notifier que la topologie réseau a été automatiquement configurée en tant que réseau linéaire. Ce message est généré par la machine de protocole RRP dont l'état est passé à LNM.

#### **0x06: NCM\_RING\_START**

Cette valeur permet de notifier que la topologie réseau a été automatiquement configurée en tant que réseau en anneau. Ce message est généré et transmis via les deux R-ports par la machine de protocole RRP dont l'état est passé à RNMP.

#### **0x07: NCM\_ACK\_RNMS**

Cette valeur est utilisée par le dispositif RNMS pour notifier que le RNMS a été correctement sélectionné. Le message NCM\_ACK\_RNMS est transmis du RNMS au RNMP via le R-port reçu du message NCM\_RING\_START ou NCM\_CHECK\_RNMS.

#### **0x08: NCM\_CHECK\_RNMS**

Cette valeur permet de demander le message NCM\_ACK\_RNMS à partir du dispositif RNMS si le dispositif RNMP souhaite vérifier l'intégrité du réseau. Le message NCM\_CHECK\_RNMS est transmis du RNMP au RNMS avec l'adresse du RNMS via le R-port1 et le R-port2.

NOTE Tous les messages du réseau RRP disposant d'une adresse de destination spécifique utilisent un R-port pour le chemin le plus court à des fins d'efficacité. Cependant, le message NCM\_CHECK\_RNMS permet également de vérifier l'intégrité du réseau RRP, en plus d'être transmis via le R-port1 et le R-port2.

### **8.3.4.3 Type of Service (ToS)**

Ce champ indique le type de service de liaison de données (DL). Une valeur de 0x00 indique un message de contrôle réseau parmi des dispositifs RRP, alors qu'une valeur de 0x01 spécifie une demande de service non confirmée parmi des utilisateurs.

### **8.3.4.4 Priority (PRI)**

Ce champ indique la priorité de la trame. Il contient la valeur du paramètre de priorité de message pour le service RRP. La priorité la plus haute est 0x03 alors que la plus basse est 0x00.

### **8.3.4.5 Validation of extension code (VoE)**

Si la trame intègre un champ d'extension, la VoE est définie sur TRUE; sinon, elle l'est sur FALSE.

## **8.4 Codage de data and pad**

### **8.4.1 Généralités**

Ce champ indique le champ de données reçu d'un utilisateur. Les messages de contrôle réseau permettent d'en transférer parmi les dispositifs RRP. Huit types de messages sont fournis pour partager les informations réseau.

### 8.4.2 Codage de FamilyReq

La trame FamilyReq est encodée comme indiqué dans le Tableau 20.

**Tableau 20 – Codage de la trame FamilyReq**

| N° | Nom de paramètre       | Type de données  | Décalage en octets | Longueur en octets | Description                                                 |
|----|------------------------|------------------|--------------------|--------------------|-------------------------------------------------------------|
| 1  | Device address         | Unsigned16       | 0                  | 2                  | Adresse de dispositif local                                 |
| 2  | Device flags           | Unsigned16       | 2                  | 2                  | Indicateurs de dispositif local                             |
| 3  | Device type            | Unsigned16       | 4                  | 2                  | Type de dispositif local                                    |
| 4  | Hop count              | Unsigned16       | 6                  | 2                  | Nombre de sauts                                             |
| 5  | Device UID             | UniqueDeviceID64 | 8                  | 8                  | Identification unique de dispositif local                   |
| 6  | Device UID for R-port1 | UniqueDeviceID64 | 16                 | 8                  | Identification unique de dispositif connecté via le R-port1 |
| 7  | Device UID for R-port2 | UniqueDeviceID64 | 24                 | 8                  | Identification unique de dispositif connecté via le R-port2 |
| 8  | MAC address            | Unsigned48       | 32                 | 6                  | Adresse MAC de dispositif local                             |
| 9  | Reserved0              | Unsigned16       | 38                 | 2                  | Réservé, – défini sur 0.                                    |
| 10 | R-port1 information    | Unsigned8        | 40                 | 1                  | Informations sur le R-port1 du dispositif local             |
| 11 | R-port2 information    | Unsigned8        | 41                 | 1                  | Informations sur le R-port2 du dispositif local             |
| 12 | Device state           | Unsigned8        | 42                 | 1                  | Etat du dispositif local                                    |
| 13 | Protocol version       | Unsigned8        | 43                 | 1                  | Version de protocole de dispositif local                    |
| 14 | Device description     | VisibleString    | 44                 | 16                 | Chaîne de description de dispositif                         |
| 15 | Reserved1              | Unsigned32       | 60                 | 4                  | Réservé, – défini sur 0.                                    |

### 8.4.3 Codage de FamilyRes

La trame FamilyRes est encodée comme indiqué dans le Tableau 21.

**Tableau 21 – Codage de la trame FamilyRes**

| N° | Nom de paramètre       | Type de données  | Décalage en octets | Longueur en octets | Description                                                 |
|----|------------------------|------------------|--------------------|--------------------|-------------------------------------------------------------|
| 1  | Device address         | Unsigned16       | 0                  | 2                  | Adresse de dispositif local                                 |
| 2  | Device flags           | Unsigned16       | 2                  | 2                  | Indicateurs de dispositif local                             |
| 3  | Device type            | Unsigned16       | 4                  | 2                  | Type de dispositif local                                    |
| 4  | Hop count              | Unsigned16       | 6                  | 2                  | Nombre de sauts                                             |
| 5  | Device UID             | UniqueDeviceID64 | 8                  | 8                  | Identification unique de dispositif local                   |
| 6  | Device UID for R-port1 | UniqueDeviceID64 | 16                 | 8                  | Identification unique de dispositif connecté via le R-port1 |
| 7  | Device UID for R-port2 | UniqueDeviceID64 | 24                 | 8                  | Identification unique de dispositif connecté via le R-port2 |
| 8  | MAC address            | Unsigned48       | 32                 | 6                  | Adresse MAC de dispositif local                             |
| 9  | Reserved0              | Unsigned16       | 38                 | 2                  | Réservé, – défini sur 0.                                    |

| N° | Nom de paramètre    | Type de données | Décalage en octets | Longueur en octets | Description                                     |
|----|---------------------|-----------------|--------------------|--------------------|-------------------------------------------------|
| 10 | R-port1 information | Unsigned8       | 40                 | 1                  | Informations sur le R-port1 du dispositif local |
| 11 | R-port2 information | Unsigned8       | 41                 | 1                  | Informations sur le R-port2 du dispositif local |
| 12 | Device state        | Unsigned8       | 42                 | 1                  | Etat du dispositif local                        |
| 13 | Protocol version    | Unsigned8       | 43                 | 1                  | Version de protocole de dispositif local        |
| 14 | Device description  | VisibleString   | 44                 | 16                 | Chaîne de description de dispositif             |
| 15 | Reserved1           | Unsigned32      | 60                 | 4                  | Réservé, – défini sur 0.                        |

#### 8.4.4 Codage de MediaLinked

La trame MediaLinked est encodée comme indiqué dans le Tableau 22.

**Tableau 22 – Codage de la trame MediaLinked**

| N° | Nom de paramètre       | Type de données  | Décalage en octets | Longueur en octets | Description                                                 |
|----|------------------------|------------------|--------------------|--------------------|-------------------------------------------------------------|
| 1  | Device address         | Unsigned16       | 0                  | 2                  | Adresse de dispositif local                                 |
| 2  | Device flags           | Unsigned16       | 2                  | 2                  | Indicateurs de dispositif local                             |
| 3  | Device type            | Unsigned16       | 4                  | 2                  | Type de dispositif local                                    |
| 4  | Hop count              | Unsigned16       | 6                  | 2                  | Nombre de sauts                                             |
| 5  | Device UID             | UniqueDeviceID64 | 8                  | 8                  | Identification unique de dispositif local                   |
| 6  | Device UID for R-port1 | UniqueDeviceID64 | 16                 | 8                  | Identification unique de dispositif connecté via le R-port1 |
| 7  | Device UID for R-port2 | UniqueDeviceID64 | 24                 | 8                  | Identification unique de dispositif connecté via le R-port2 |
| 8  | MAC address            | Unsigned48       | 32                 | 6                  | Adresse MAC de dispositif local                             |
| 9  | Reserved0              | Unsigned16       | 38                 | 2                  | Réservé, – défini sur 0.                                    |
| 10 | R-port1 information    | Unsigned8        | 40                 | 1                  | Informations sur le R-port1 du dispositif local             |
| 11 | R-port2 information    | Unsigned8        | 41                 | 1                  | Informations sur le R-port2 du dispositif local             |
| 12 | Device state           | Unsigned8        | 42                 | 1                  | Etat du dispositif local                                    |
| 13 | Protocol version       | Unsigned8        | 43                 | 1                  | Version de protocole de dispositif local                    |
| 14 | Device description     | VisibleString    | 44                 | 16                 | Chaîne de description de dispositif                         |
| 15 | Reserved1              | Unsigned32       | 60                 | 4                  | Réservé, – défini sur 0.                                    |

#### 8.4.5 Codage de AdvThis

La trame AdvThis est encodée comme indiqué dans le Tableau 23.