

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – High availability automation networks –
Part 2: Media Redundancy Protocol (MRP)**

**Réseaux de communication industriels – Réseaux d'automatisme à haute
disponibilité –
Partie 2: Protocole de redondance du support (MRP)**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2016 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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

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

About the IEC

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

About IEC publications

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

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

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

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

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

A propos de l'IEC

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

A propos des publications IEC

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

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

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

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 15 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

65 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

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

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – High availability automation networks –
Part 2: Media Redundancy Protocol (MRP)**

**Réseaux de communication industriels – Réseaux d'automatisme à haute
disponibilité –
Partie 2: Protocole de redondance du support (MRP)**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040, 35.040

ISBN 978-2-8322-3149-4

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

CONTENTS

FOREWORD.....	7
INTRODUCTION.....	9
1 Scope.....	11
2 Normative references.....	11
3 Terms, definitions, abbreviations, acronyms, and conventions	12
3.1 Terms and definitions	12
3.2 Abbreviations and acronyms.....	12
3.3 Conventions.....	12
4 MRP Overview.....	12
5 MRP Media redundancy behavior.....	16
5.1 General.....	16
5.2 Ring ports	16
5.3 Media Redundancy Manager (MRM).....	17
5.4 Media Redundancy Client (MRC).....	19
5.5 Redundancy domain.....	19
5.6 Media Link Check.....	19
5.7 Application of the Continuity Check protocol	19
5.7.1 General	19
5.7.2 Continuity Check Message Interval.....	20
5.7.3 Maintenance Domain Level	20
5.7.4 Maintenance Association ID (MAID).....	20
5.7.5 Maintenance Association End Point ID (MEPID).....	20
5.7.6 Sender ID TLV	20
5.7.7 Port Status TLV	21
5.7.8 Interface Status TLV	21
5.8 Usage with diagnosis and alarms.....	21
5.9 Ring diagnosis	21
5.10 Multiple MRM in a single ring: Manager voting option	21
5.10.1 General	21
5.10.2 Basic principle of the manager voting process	22
5.10.3 The manager voting process	23
5.11 BLOCKED not supported (Option).....	25
5.12 Interconnection port	25
5.13 Media redundancy Interconnection Manager (MIM)	26
5.14 Media redundancy Interconnection Client (MIC)	29
5.15 Interconnection domain	29
5.16 Interconnection diagnosis.....	30
6 MRP Class specification	30
6.1 General.....	30
6.2 Template.....	30
6.2.1 Media redundancy template.....	30
6.2.2 Media redundancy Interconnection template	31
6.3 Attributes	32
7 MRP Service specification	36
7.1 Start MRM	36

7.2	Stop MRM.....	38
7.3	State Change.....	38
7.4	Start MRC.....	39
7.5	Stop MRC.....	41
7.6	Read MRM.....	41
7.7	Read MRC.....	43
7.8	Start MIM.....	45
7.9	Stop MIM.....	46
7.10	Interconnection State Change.....	47
7.11	Start MIC.....	48
7.12	Stop MIC.....	49
7.13	Read MIM.....	50
7.14	Read MIC.....	52
8	MRP protocol specification.....	54
8.1	PDU description.....	54
8.1.1	Basic data types.....	54
8.1.2	DLPDU abstract syntax reference.....	54
8.1.3	Coding of the DLPDU field SourceAddress.....	54
8.1.4	Coding of the DLPDU field DestinationAddress.....	55
8.1.5	Coding of the field TagControlInformation.....	55
8.1.6	Coding of the field LT.....	56
8.1.7	MRP APDU abstract syntax.....	56
8.1.8	Coding of the field MRP_TLVHeader.....	57
8.1.9	Coding of the field MRP_SubTLVHeader.....	58
8.1.10	Coding of the field MRP_Ed1Type and MRP_Ed1ManufacturerData.....	58
8.1.11	Coding of the field MRP_Version.....	59
8.1.12	Coding of the field MRP_SequenceID.....	59
8.1.13	Coding of the field MRP_SA.....	59
8.1.14	Coding of the field MRP_OtherMRMSA.....	59
8.1.15	Coding of the field MRP_Prio.....	60
8.1.16	Coding of the field MRP_OtherMRMPrio.....	60
8.1.17	Coding of the field MRP_PortRole.....	60
8.1.18	Coding of the field MRP_RingState.....	60
8.1.19	Coding of the field MRP_Interval.....	61
8.1.20	Coding of the field MRP_Transition.....	61
8.1.21	Coding of the field MRP_TimeStamp.....	61
8.1.22	Coding of the field MRP_Blocked.....	61
8.1.23	Coding of the field MRP_ManufacturerOUI.....	62
8.1.24	Coding of the field MRP_IECOUI.....	62
8.1.25	Coding of the field MRP_ManufacturerData.....	62
8.1.26	Coding of the field MRP_DomainUUID.....	62
8.1.27	Coding of the field MRP_InState.....	62
8.1.28	Coding of the field MRP_InID.....	63
8.2	Protocol machines.....	63
8.2.1	MRM protocol machine.....	63
8.2.2	MRC protocol machine.....	74
8.2.3	MRA protocol machine.....	80
8.2.4	MRA, MRM and MRC functions.....	100
8.2.5	FDB clear timer.....	105

8.2.6	Topology change timer	105
8.2.7	MIM protocol machine	106
8.2.8	MIC protocol machine	115
8.2.9	MIM and MIC functions.....	123
8.2.10	Interconnection Topology Change timer.....	127
8.2.11	Interconnection Link Status Poll timer.....	127
9	MRP installation, configuration and repair	128
9.1	Ring port and Interconnection port parameters.....	128
9.2	Ring topology parameters.....	128
9.3	MRM parameters.....	128
9.4	MRC parameters and constraints.....	129
9.5	MRA compatibility to earlier Automanager protocol version	129
9.6	Interconnection topology parameters	130
9.7	MIM parameters	130
9.8	MIC parameters and constraints	130
9.9	Calculation of MRP ring recovery time	131
9.9.1	Overview	131
9.9.2	Deduction of formula	131
9.9.3	Worst case calculation for recovery time of 10 ms.....	133
9.9.4	Worst case calculation for 50 devices	134
9.10	Calculation of MRP Automanager voting time.....	134
10	MRP Management Information Base (MIB)	134
10.1	General.....	134
10.2	MRP MIB with a monitoring view.....	134
10.3	MRP MIB with a management and monitoring view	147
Annex A (normative)	Optional earlier version of the Automanager protocol.....	162
Bibliography		163
Figure 1	Two MRP rings redundantly connected via MRP Interconnection	14
Figure 2	MRP stack	16
Figure 3	MRP ring topology with one manager and clients	17
Figure 4	MRP open ring with MRM	18
Figure 5	MRP ring with MRA at network startup	22
Figure 6	MRP ring after the manager voting process.....	22
Figure 7	Manager voting process	24
Figure 8	MRA located outside the MRP ring.....	25
Figure 9	MRP Interconnection topology	27
Figure 10	MRP ring interconnection open	28
Figure 11	MRP protocol machine for MRM.....	63
Figure 12	MRP protocol machine for MRC	74
Figure 13	MRP protocol machine for MRA	81
Figure 14	MRP protocol machine for MIM in RC-mode and LC-mode.....	107
Figure 15	MRP protocol machine for MIC in RC-mode and LC-mode	116
Table 1	Patent information	9
Table 2	Coding of the Maintenance Association ID (MAID).....	20

Table 3 – MRP Start MRM	36
Table 4 – MRP Stop MRM.....	38
Table 5 – MRP Change State	39
Table 6 – MRP Start MRC.....	40
Table 7 – MRP Stop MRC	41
Table 8 – MRP Read MRM.....	42
Table 9 – MRP Read MRC	44
Table 10 – MRP Start MIM.....	45
Table 11 – MRP Stop MIM	47
Table 12 – MRP Interconnection Change State.....	47
Table 13 – MRP Start MIC	48
Table 14 – MRP Stop MIC.....	50
Table 15 – MRP Read MIM	51
Table 16 – MRP Read MIC.....	53
Table 17 – MRP DLPDU syntax for ISO/IEC/IEEE 8802-3 (IEEE 802.3)	54
Table 18 – MRP OUI.....	55
Table 19 – MRP MulticastMACAddress	55
Table 20 – MRP TagControlInformation.Priority field.....	56
Table 21 – MRP LT field	56
Table 22 – MRP APDU syntax.....	56
Table 23 – MRP Substitutions	57
Table 24 – MRP_TLVHeader.Type	58
Table 25 – MRP_SubTLVHeader.Type	58
Table 26 – MRP_Ed1Type and MRP_Ed1ManufacturerData	59
Table 27 – MRP_Ed1Type and MRP_Ed1ManufacturerData	59
Table 28 – MRP_Version	59
Table 29 – Coding of the field MRP_OtherMRMSA	60
Table 30 – MRP_Prio.....	60
Table 31 – Coding of the field MRP_OtherMRMPrio.....	60
Table 32 – MRP_PortRole.....	60
Table 33 – MRP_RingState.....	61
Table 34 – MRP_Interval.....	61
Table 35 – MRP_Transition.....	61
Table 36 – MRP_TimeStamp.....	61
Table 37 – MRP_Blocked.....	62
Table 38 – MRP_DomainUUID	62
Table 39 – MRP_InState	62
Table 40 – MRP Local variables of MRM protocol machine	65
Table 41 – MRM State machine	66
Table 42 – MRP Local variables of MRC protocol machine.....	75
Table 43 – MRC state machine	76
Table 44 – MRP local variables of MRA protocol machine.....	82
Table 45 – MRA state machine.....	83

Table 46 – MRP functions and macros	101
Table 47 – MRP FDB clear timer	105
Table 48 – MRP topology change timer	106
Table 49 – MRP Local variables of MIM protocol machine.....	108
Table 50 – MIM State machine for LC-mode	109
Table 51 – MIM State machine for RC-mode	112
Table 52 – MRP Local variables of MIC protocol machine	117
Table 53 – MIC State machine for LC-mode	118
Table 54 – MIC State machine for RC-mode	121
Table 55 – MRP Interconnection functions.....	124
Table 56 – MRP Interconnection topology change timer.....	127
Table 57 – MRP Interconnection link status poll timer	128
Table 58 – MRP Network/Connection parameters	128
Table 59 – MRP MRM parameters.....	129
Table 60 – MRP MRC parameters	129
Table 61 – MRP MIM parameters	130
Table 62 – MRP MIC parameters	131
Table A.1 – Compatible mode MRP_Option for MRP_Test Substitutions	162
Table A.2 – Compatible mode MRP_Option frames MRP_TestMgrNack and MRP_TestPropagate Substitutions	162

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS –
HIGH AVAILABILITY AUTOMATION NETWORKS –

Part 2: Media Redundancy Protocol (MRP)

FOREWORD

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

International Standard IEC 62439-2 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2010. This edition constitutes a technical revision.

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

- adding a protocol extension to select the media redundancy manager automatically;
- adding a protocol to redundantly connect media redundancy protocol rings.

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

FDIS	Report on voting
65C/834/FDIS	65C/841/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 publication has been drafted in accordance with ISO/IEC Directives, Part 2.

This International Standard is to be read in conjunction with IEC 62439-1.

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

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

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

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.

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/IEEE 8802-3 (IEEE 802.3) compatible, reliable data communication, and preserve determinism of real-time data communication. In cases of fault, removal, and insertion of a component, they provide deterministic recovery times.

These protocols retain fully the typical Ethernet communication capabilities as used in the office world, so that the software involved remains applicable.

The market is in need of several network solutions, each with different performance characteristics and functional capabilities, matching diverse application requirements. These solutions support different redundancy topologies and mechanisms which are introduced in IEC 62439-1 and specified in the other Parts of the IEC 62439 series. IEC 62439-1 also distinguishes between the different solutions, giving guidance to the user.

The IEC 62439 series follows the general structure and terms of the 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 a patent concerning ring protocol given in Clause 5. Table 1 gives an overview of the relevant patents.

Table 1 – Patent information

No.	Status	Country	Granted Patent Number or Application Number (if pending)	Title
1	granted	US	US 6430151	Local networking with redundancy properties having a redundancy manager
	granted	CA	CA 2323429	
	granted	CN	CN 117195	
	granted	NO	NO 330908	
	granted	EP (AT, BE, CH, DE, DK, ES, FR, GB, IT, NL, SE)	EP 1062787	

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

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

Siemens Aktiengesellschaft

Otto-Hahn-Ring 6

Munich 81739

Germany

and

Hirschmann Automation and Control GmbH

Stuttgarter Strasse 45-51

Neckartenzlingen 72654

Germany

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

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

INDUSTRIAL COMMUNICATION NETWORKS – HIGH AVAILABILITY AUTOMATION NETWORKS –

Part 2: Media Redundancy Protocol (MRP)

1 Scope

The IEC 62439 series is applicable to high-availability automation networks based on the ISO/IEC/IEEE 8802-3 (IEEE 802.3) (Ethernet) technology.

This part of the IEC 62439 series specifies a recovery protocol based on a ring topology, designed to react deterministically on a single failure of an inter-switch link or switch in the network, under the control of a dedicated media redundancy manager node.

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 61158-6-10:2014, *Industrial communication networks – Fieldbus specifications – Part 6-10: Application layer protocol specification – Type 10 elements*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

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

IEC 62439-1:2010/AMD1:2012¹

ISO/IEC 10164-1, *Information technology – Open Systems Interconnection – Systems Management: Object Management Function*

ISO/IEC/IEEE 8802-3 *Standard for Ethernet*

IEEE 802.1D:2004, *IEEE Standard for Local and metropolitan area networks – Media Access Control (MAC) Bridges*

IEEE 802.1Q:2011, *IEEE Standard for Local and metropolitan area networks – Media Access Control (MAC) Bridges and Virtual Bridge Local Area Network*

¹ A consolidated version of this publication exists, comprising IEC 62439-1:2010 and IEC 62439-1:2010/AMD1:2012.

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, as well as the following, apply.

3.1.1

interconnection port

port of a switch that is used to interconnect two ring topologies

3.1.2

interconnection topology

topology in which two ring topologies are connected

3.2 Abbreviations and acronyms

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

ASE	Application Service Element
CCM	Continuity Check Messages
CFM	Connectivity Fault Management
CFM-MRP	Connectivity Fault Management for MRP
CPU	Central Processing Unit
FDB	Filtering Database
LC-mode	Link Check mode
LSB	Least Significant Bit
MAID	Maintenance Association ID
MD	Maintenance Domain
MD Level	Maintenance Domain Level
MEP	Maintenance End Point
MEPID	Maintenance Association End Point ID
MIB	Management Information Base
MRA	Media Redundancy Automanager
MRC	Media Redundancy Client
MRM	Media Redundancy Manager
MRP	Media Redundancy Protocol
MIC	Media Redundancy Interconnection Client
MIM	Media Redundancy Interconnection Manager
RC-mode	Ring Check mode

3.3 Conventions

This document follows the conventions defined in IEC 62439-1.

4 MRP Overview

This document specifies the Media Redundancy Protocol (MRP), a recovery protocol based on a ring topology, and the redundant interconnection of MRP rings via the Media Redundancy Interconnection Protocol as shown in Figure 1.

MRP is designed to react deterministically on a single failure of an inter-switch link or switch in the ring or interconnection topology.

MRP is based on functions of ISO/IEC/IEEE 8802-3 (IEEE 802.3) and IEEE 802.1Q including the Filtering Database (FDB) and is located between the Data Link Layer and Application Layer (see Figure 2).

NOTE Layering is assumed to be according to IEC 61158-1.

A MRP compliant network shall have a ring topology with multiple nodes.

One of the nodes has the role of a Media Redundancy Manager (MRM). The function of the MRM is to observe and to control the ring topology in order to react on network faults. The MRM does this by sending frames on one ring port over the ring and receiving them from the ring over its other ring port, and vice-versa in the other direction.

The other nodes in the ring have the role of Media Redundancy Clients (MRC). An MRC reacts on received reconfiguration frames from the MRM and can detect and signal link changes on its ring ports.

Certain nodes or all nodes in the ring may also start as a Media Redundancy Automanager (MRA). MRAs select one MRM among each other by using a voting protocol. The remaining MRAs transition to the role MRC.

Each node in the ring is able to detect the failure or recovery of an inter-switch link or the failure or recovery of a neighboring node.

To redundantly connect two MRP rings, two nodes of each ring are assigned additional roles.

One of the nodes has the role of a media redundancy interconnection manager (MIM), in addition to the role of a MRC or MRM. The function of the MIM is to observe and to control the redundant interconnection topology in order to react on interconnection faults. To cover a maximum of applications, two detection methods are provided by this international standard. The MIM can observe the interconnection topology by either reacting directly on interconnection port link change notification messages (LC-mode) or by sending test frames on the interconnection port over the connected rings and receiving them over its ring ports, and vice-versa in the other direction (RC-mode).

The other three nodes in the interconnection topology have the role of media redundancy interconnection clients (MIC), in addition to the role of a MRC or MRM. The MIC reacts on received reconfiguration frames from the MIM, it can detect and signal link changes of its interconnection port, and it can issue link change notification messages.

As described in the course of this international standard, LC-mode has the advantage of restricting the interconnection test frame load only to the interconnection links, whereas RC-mode has the advantage of deloading the MICs from interconnection test frame processing. The selection of the mode has to be made in accordance with the requirements of the application and is outside the scope of this international standard.

The vendor may implement only the LC-mode or only the RC-mode or both modes. The MIM and all MICs of one MRP interconnection topology shall be configured to the same mode.

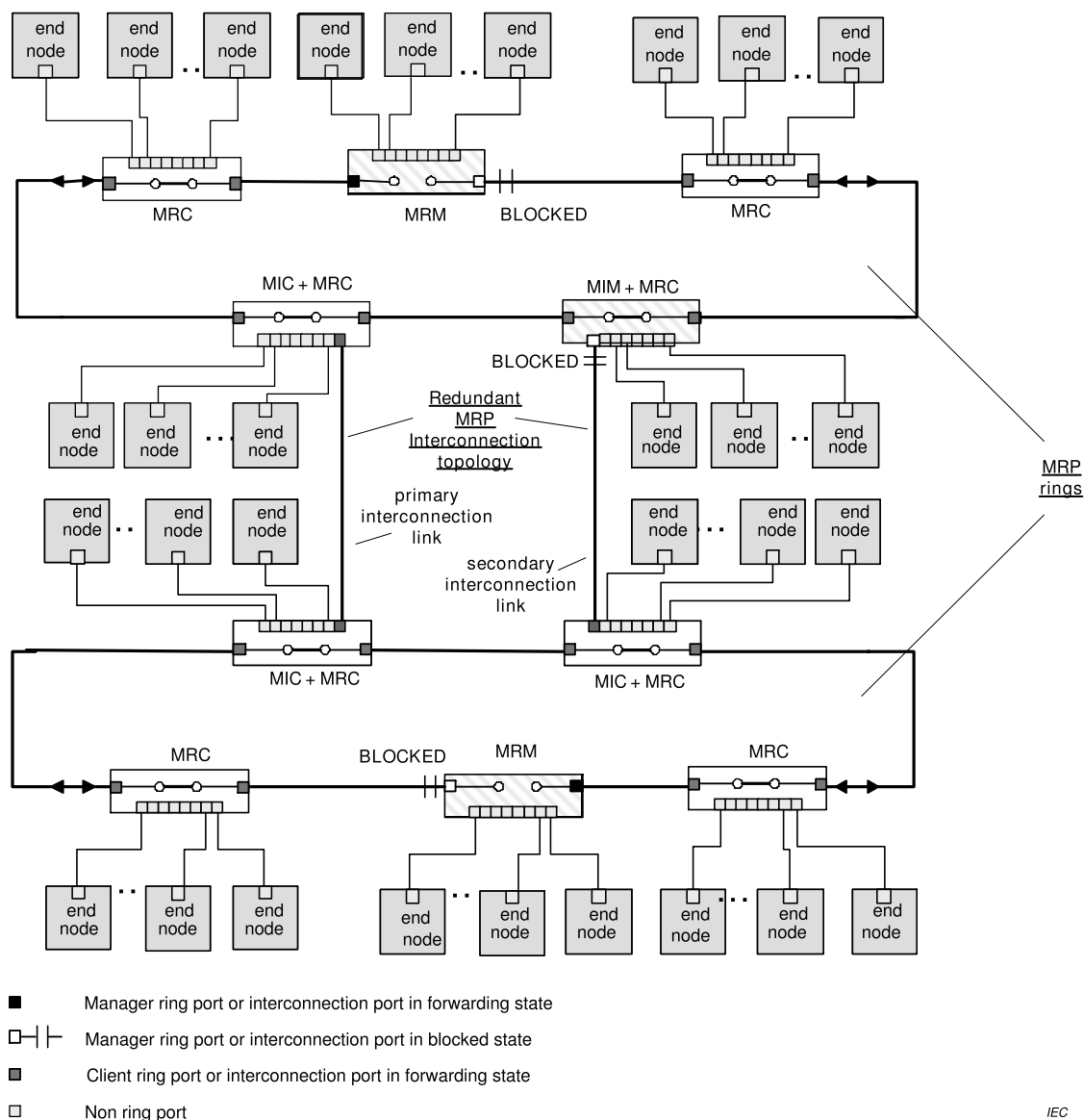


Figure 1 – Two MRP rings redundantly connected via MRP Interconnection

Each MRP compliant node requires an IEEE 802.1 compliant switch element with two ring ports connected to one ring and shall have the ability to perform at least as one of the following:

- media redundancy manager (MRM); or
- media redundancy client (MRC); or
- both MRM or MRC (but both roles shall not be active at the same time for the same ring).

Each Media Redundancy Interconnection compliant node requires an IEEE 802.1 compliant switch element with two ring ports connected to one ring, and with one interconnection port connected to another MRP ring and shall have the ability to perform at least as one of the following:

- both MRC and media redundancy interconnection manager (MIM) (both roles shall be active at the same time); or
- both MRM and MIM (both roles shall be active at the same time); or
- both MRC and media redundancy interconnection client (MIC) (both roles shall be active at the same time); or

- both MRM and MIC (both roles shall be active at the same time).

More than two MRP rings may be redundantly connected by applying the redundant interconnection topology multiple times from one ring to other rings. This leads to rings containing multiple nodes configured to MIM and multiple nodes configured to MIC. A redundant connection of more than two rings may also be achieved by two nodes of a ring containing multiple instances of MIMs or MICs.

An MRP compliant node may implement multiple instances of the MRC, MRM, MRA, MIM or MIC functionality. In this case it shall obey the following configuration rules:

- if one or more MIM or MIC instances are active on one node, then there shall be only one MRC or one MRM or one MRA instance active on this node at the same time;
- if more than one MRC or MRM or MRA instances are active on one node, then no MIC or MIM shall be active on this node at the same time;
- if more than one MRC or MRM or MRA instances are active on one node, thereby connecting the related rings connected to the ring ports of this node, then there shall be no other node in the network connecting the same rings over its MRC or MRM or MRA instances.

The following definitions and chapters with MRP and MRP Interconnection definitions always apply to one MRP instance or one MRP Interconnection instance per compliant node. Multiple activation of MRP instances or MRP Interconnection instances, according to the configuration rules stated above, might demand additional requirements to correct frame processing and frame forwarding, which are out of the scope of this document.

NOTE Additional interconnection ports can be used to connect to additional rings.

The MRP consists of a service and a protocol entity, see stack model in Figure 2.

The service entity specifies, in an abstract way, the externally visible service provided by the Data Link Layer in terms of:

- primitive actions and events of the service;
- parameters associated with each primitive action and event, and the form which they take, and;
- interrelationship between these actions and events, and their valid sequences.

MRP defines the services provided to:

- the Application Layer at the boundary between the Application Layer and the Data Link Layer; and
- the MRP Management at the boundary between the Data Link Layer and the MRP Management.

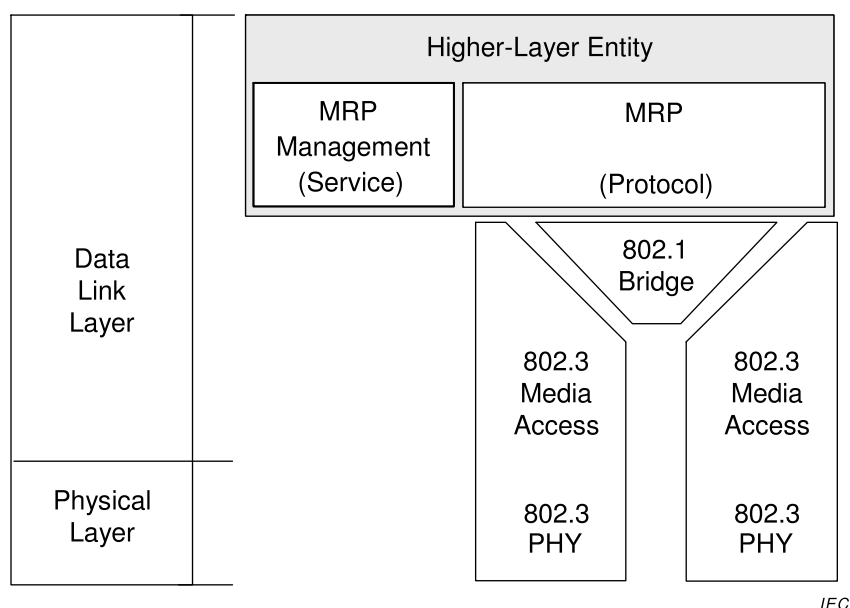


Figure 2 – MRP stack

5 MRP Media redundancy behavior

5.1 General

The following definitions for Media redundancy protocol roles always refer to one instance of the MRP role.

5.2 Ring ports

The MRM and the MRC shall have two ring ports.

The MRM and MRC shall be able to detect the failure or recovery of a link on a ring port with mechanisms based on ISO/IEC/IEEE 8802-3 (IEEE 802.3).

The MRM and MRC shall not forward MRP_Test frames, MRP_TopologyChange frames, and MRP_LinkChange frames to non-ring ports.

NOTE Ring ports are expected to behave as if STP, RSTP or MSTP is disabled.

A ring port shall take one of the following port states:

- DISABLED

All frames shall be dropped.

- BLOCKED

All frames shall be dropped except the following:

- MRP_TopologyChange frames, MRP_Test frames and MRP_TestMgrNAck frames;
- MRP_LinkChange frames and MRP_TestPropagate from an MRC;
- MRP_Option frames;
- MRP_InTopologyChange and MRP_InLinkStatusPoll from an MIM;
- MRP_InLinkChange frames from an MIC;
- frames specified in IEEE 802.1D-2004 Table 7-10 to pass ports in “Discarding” state (e.g. LLDP, IEEE 1588-2008 PTP peer to peer frames);

- frames only produced or consumed by the higher layer entities of this node and never forwarded.

– FORWARDING

All frames shall be passed through according to the forwarding behavior of IEEE 802.1D.

NOTE IEEE 802.1D refers to the port state corresponding to BLOCKED as “Discarding”.

5.3 Media Redundancy Manager (MRM)

One ring port of the MRM shall be connected to a ring port of an MRC. The other ring port of that MRC shall be connected to a ring port of another MRC or to the second ring port of the MRM, thereby forming a ring topology as shown in Figure 3.

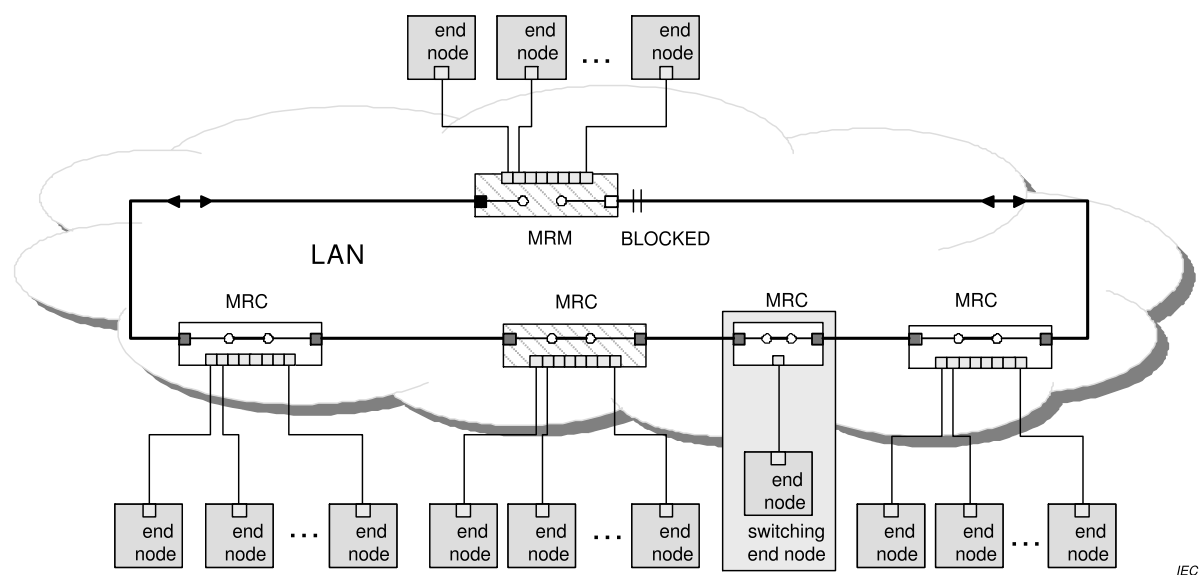


Figure 3 – MRP ring topology with one manager and clients

The MRM shall control the ring state by:

- sending MRP_Test frames at a configured time period in both directions of the ring;
- setting one ring port to FORWARDING state and the other ring port to BLOCKED state if it receives its own MRP_Test frames (this means that the ring is closed, see Figure 3);
- setting both ring ports to FORWARDING state if it does not receive its own MRP_Test frames on both ring ports within a configured time according to MRP_TSTdefaultT, MRP_TSTshortT and MRP_TSTNRmax in Table 59 (this means that the ring is open, see Figure 4).

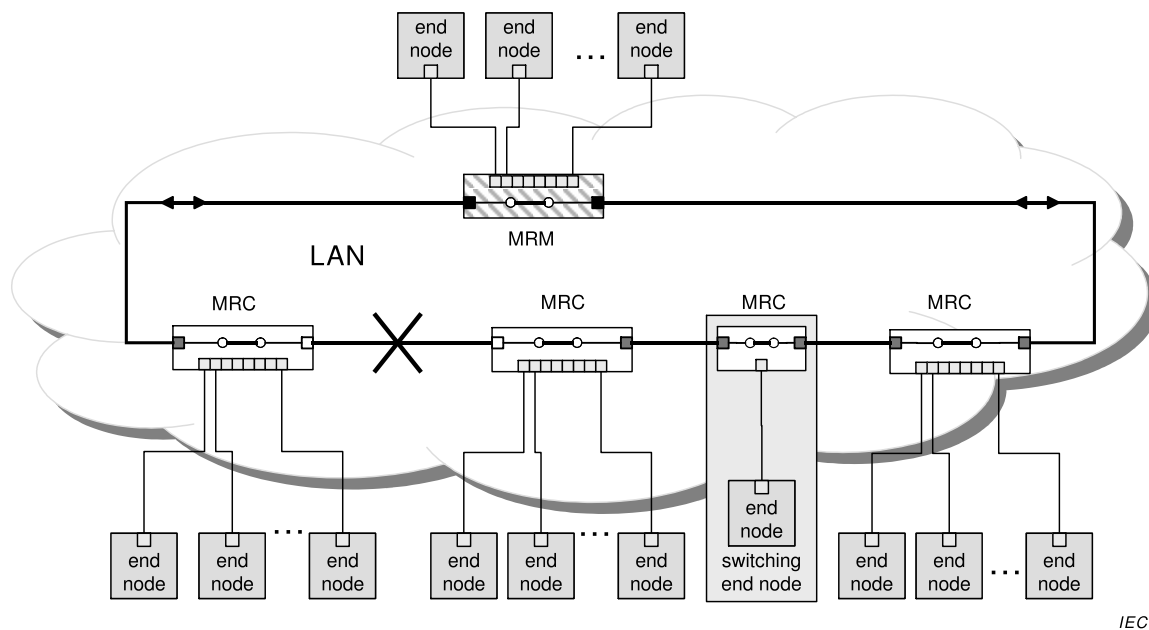


Figure 4 – MRP open ring with MRM

The following mechanism supports synchronization between MRM and MRC in ring topology changes.

The MRM shall indicate changes in the ring state to the MRCs by means of MRP_TopologyChange frames.

The MRM shall not forward MRP specific frames (MRP_Test frames, MRP_TopologyChange frames, MRP_LinkChange frames) between its ring ports.

If the MRM receives an MRP_LinkUp or MRP_LinkDown frame, then the MRM shall reduce its test monitoring time according to Table 59 to accelerate the detection of the open ring. When the open ring is detected then the MRM shall set one ring port to forwarding state and shall send the MRP_TopologyChange frames through both its ring ports.

If parameter MRP_REACT_ON_LINK_CHANGE is set, see Table 41, the MRM shall, on receiving an MRP_LinkDown frame, immediately set one ring port to forwarding state and shall then send the MRP_TopologyChange frames through its ring ports.

The MRM shall send to the MRCs an MRP_TopologyChange frame containing the delay, after which the ring topology change will be performed. The parameter carrying this delay is called MRP_Interval. When this time has expired, all MRCs shall clear their Filtering Database (FDB) information for their ring ports.

Each MRC shall send the configured delay in MRP_Interval to the MRM in the MRP_LinkUp and MRP_LinkDown frames to tell the MRM after which time the MRC will change its port state from BLOCKED to FORWARDING (MRP_LinkUp frame) or from FORWARDING to BLOCKED (MRP_LinkDown frame).

Measures shall be included to prevent the MRM from remaining stuck in the closed state in case of node failure.

The MRM shall not forward MRP_InTest frames, MRP_InLinkChange frames, MRP_InLinkStatusPoll frames and MRP_InTopologyChange frames between its ring ports, if one of its ring port is in the port state BLOCKED.

The MRM shall process MRP_InTopologyChange frames. It shall clear its FDB if requested by an MRP_InTopologyChange frame in a given time interval MRP_Interval. It shall send an MRP_TopologyChange frame if requested by the MRP_InTopologyChange frame, in order to ensure that all MRC, MIM and MIC in the ring domain clear their FDB at the same time.

5.4 Media Redundancy Client (MRC)

Each MRC shall forward MRP_Test frames received on one ring port to the other ring port and vice versa.

If the MRC detects a failure or recovery of a ring port link, the MRC may optionally notify the change by sending MRP_LinkChange frames through both of its ring ports. Each MRC shall forward MRP_LinkChange frames received on one ring port to the other ring port and vice versa.

Each MRC shall forward MRP_TopologyChange frames received on one ring port to the other ring port and vice versa. Each MRC shall process these frames. It shall clear its FDB if requested by an MRP_TopologyChange frame in a given time interval MRP_Interval.

Each MRC shall forward MRP_InTest frames, MRP_InLinkChange frames, MRP_InTopologyChange frames and MRP_InLinkStatusPoll frames between its ring ports, if no MIM and no MIC is active on the node at the same time.

5.5 Redundancy domain

The redundancy domain represents a ring. By default, all MRM and MRCs belong to the default domain. A unique domain ID can be allocated to all ring nodes of one ring as a key attribute. This is especially essential to distinguish frames if a device is member of multiple rings by implementing multiple instances of MRMs or MRCs. A node shall assign exactly two unique ring ports per redundancy domain.

NOTE A device can have other ports than the two assigned to MRP. These other ports are not influenced by MRP.

5.6 Media Link Check

To enhance the quality of the link detection of certain ring or interconnection ports, a media link check protocol may be used. It shall be used for LC-mode.

The Continuity Check protocol, defined in the Connectivity Fault Management (CFM) protocol suite within IEEE 802.1Q-2011 and further specified in this document under the name Connectivity Fault Management for MRP (CFM-MRP), shall be supported as media link check protocol.

CFM-MRP shall be used at the interconnection ports of the MIM and MICs when operating the ring interconnection in LC-mode. CFM-MRP may be used at the interconnection ports if operating the ring interconnection in RC-mode.

CFM-MRP may also be used at the ring ports of MRP devices.

5.7 Application of the Continuity Check protocol

5.7.1 General

When applying the CFM-MRP protocol according to the specification of the Continuity Check protocol as defined in IEEE 802.1Q-2011 to check the links between ring- or interconnection ports, then the ring- and interconnection ports represent Maintenance End Points (MEP) belonging to a Maintenance Domain (MD) and a Maintenance Association (MA) in the sense of IEEE 802.1Q-2011.

An MRP device, applying the CFM-MRP protocol on its interconnection ports or ring ports, shall internally produce a MAUTypeChangeInd event when the Continuity Check receiver instance on its MRP port has detected a remote Maintenance End Point (MEP) error by detecting a lifetime violation of the Continuity Check messages (CCM) from the linked remote MRP devices MEP CFM-MRP initiator.

5.7.2 Continuity Check Message Interval

The CCM Interval to be used shall be either 3,33 ms or 10 ms according to IEEE 802.1Q-2011.

5.7.3 Maintenance Domain Level

The Maintenance Domain Level (MD Level) to be used between two ring- or interconnection ports of two MRP devices shall be level 0 for lowest level physical connections. Thereby also the fixed CCM Multicast MAC Address 01-80-C2-00-00-30 is assigned according to IEEE 802.1Q-2011.

5.7.4 Maintenance Association ID (MAID)

The Maintenance Association ID (MAID) according to IEEE 802.1Q-2011 to be used may be as specified in Table 2

Table 2 – Coding of the Maintenance Association ID (MAID)

Subfield	Value
Maintenance Domain Name Format (Character String)	4
Maintenance Domain Name Length	Variable
Maintenance Domain Name	Character string containing: – for ring ports: the MRP Domain name – for interconnection ports: The Interconnection name plus either a string extension .prm for primary link or .sec for secondary link or .na during startup.^a
Short MA Name Format (2-octet integer)	3
Short MA Name Length	2
Short MA Name	two Octets containing: – for ring ports: 0x0000 – for interconnection ports: Interconnection ID MRP_InID.
^a For a CCM MEP initiator to know whether it is sending on a secondary link is given by the fact that the MEP initiator either residing on an MIM or an MEP receiver receives CCM frames from an MEP initiator with the MIM role specified in the IEC specific TLV. Otherwise it is sending on a primary link.	

5.7.5 Maintenance Association End Point ID (MEPID)

The content of the 2-octet Maintenance Association End Point ID is out of the scope of this document.

5.7.6 Sender ID TLV

The use of the Sender ID TLV according to IEEE 802.1Q is optional for MRP.

5.7.7 Port Status TLV

The use of the Port Status TLV according to IEEE 802.1Q is optional for MRP.

5.7.8 Interface Status TLV

The use of the Interface Status TLV according to IEEE 802.1Q is optional for MRP.

5.8 Usage with diagnosis and alarms

If the attribute Check Media Redundancy has the value TRUE, media redundancy events shall cause diagnosis events and alarm notifications.

5.9 Ring diagnosis

In a redundancy domain the following diagnosis events handling shall be implemented by each MRM:

- If a device is configured as MRM, but not operating in the manager role, it shall signal a “MANAGER_ROLE_FAIL” diagnosis event and suspend reporting of all other media redundancy diagnosis events while not in the manager role.
- If a device is operating in manager role and this device detects another active MRM, it shall signal the “MULTIPLE MANAGERS” event. This event can occur concurrently with the ring state event “RING_OPEN”.
- If a device is operating in manager role and detects an open ring, it shall signal the “RING_OPEN” event.

These events shall be signaled by using the State Change service (see 7.3).

NOTE The presence of MRP_Test frames enables the checking of the existence of an MRM.

5.10 Multiple MRM in a single ring: Manager voting option

5.10.1 General

There shall be only one active MRM in the ring even if several nodes have this ability.

NOTE 1 Multiple active MRMs cause the ring to divide itself into several segments.

As an alternative to manual MRP-role configuration, in case of more than one node having the ability to become an MRM in the ring, an enhanced manager voting option, specified in this International Standard, may be used to decide which of these nodes shall finally become the MRM, while the other nodes take over the MRC role as shown in Figure 5 and Figure 6. To this effect, the nodes with the MRM ability have different priorities that shall be conveyed in the MRP_Prio field of the MRP_Test frame.

If an option for multiple MRM in a single ring is used, then all MRM in the ring shall support the same manager voting option, as defined in this document.

MRM supporting the manager voting option are named Media Redundancy Automanager or MRA.

The MRA role is an MRP role which comprises both the MRP roles MRM and MRC. Additionally the MRA has the capability to transition at any point in time, by virtue of the voting protocol, from its initial operational role manager (MRA) to the operational role client (MRC), and back again, in coordination with possible other MRAs in the ring.

NOTE 2 To emphasize the principle, the following explanations use the expression MRM for the remaining MRA managing the ring, although the device continues to be MRA, always capable to vote again.

Figure 5 shows an MRP ring at network startup with two devices in the administrative MRA role.

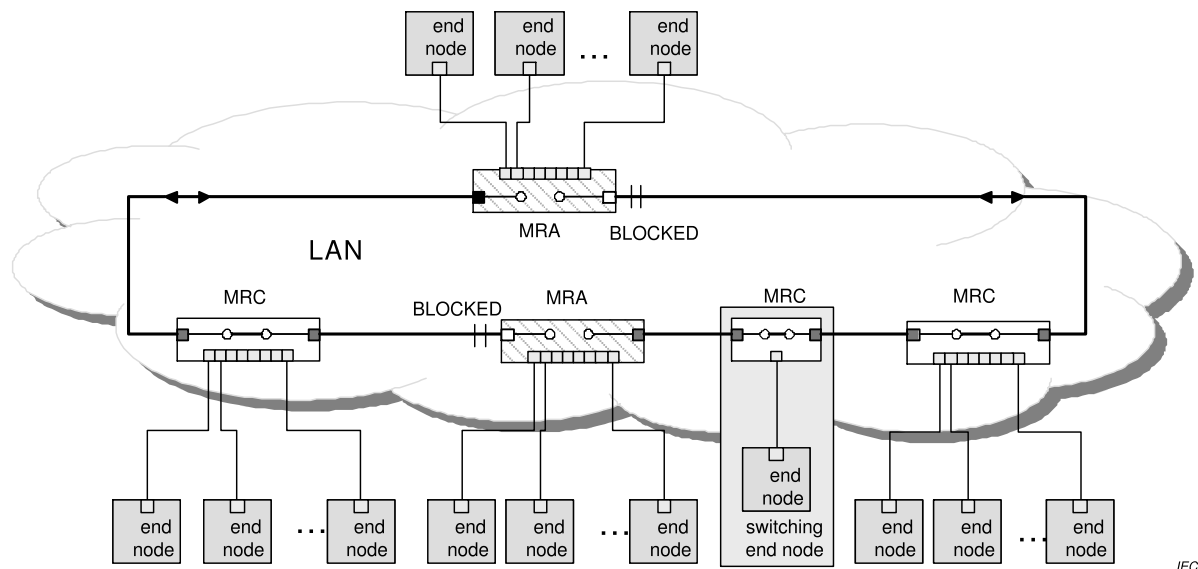


Figure 5 – MRP ring with MRA at network startup

Figure 6 shows the same network ring after the manager voting process has taken place. The MRA transitioned to the MRP roles MRM and MRC.

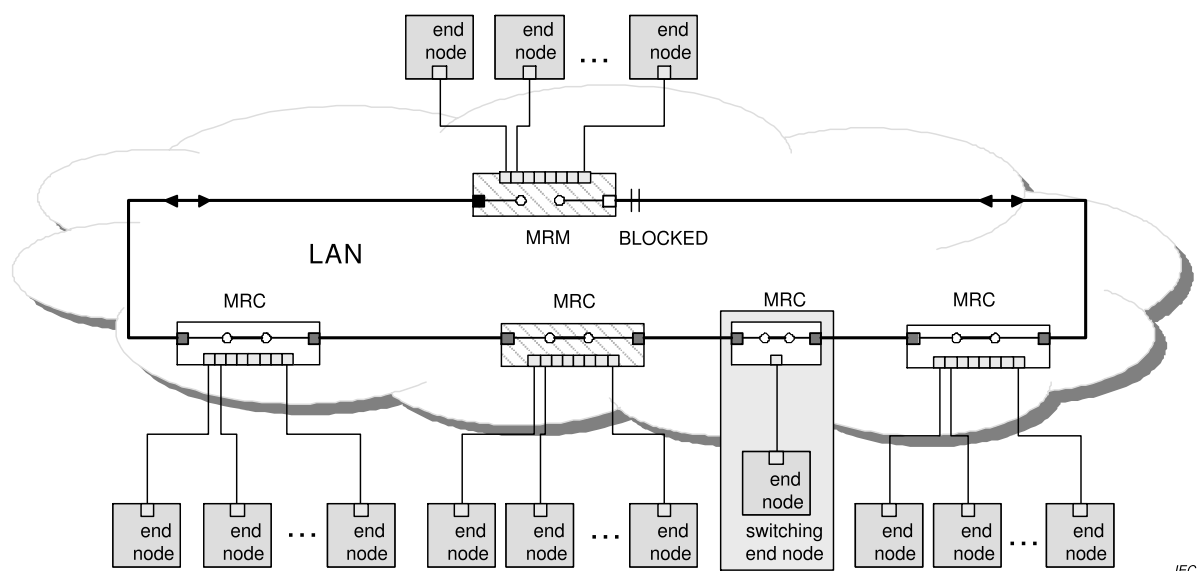


Figure 6 – MRP ring after the manager voting process

5.10.2 Basic principle of the manager voting process

The manager voting is an optional process only for devices which are capable of taking over the MRM role within MRP.

The manager voting phase of MRP shall take place right after device startup, and before the device starts to take one of the possible MRP-roles MRM or MRC.

The basic functionality of the manager voting process shall be as follows:

- Each network node of the ring with MRM-capability and manager voting option support, thereby being a so-called “Manager (Auto)” or short “MRA”, may be set to the MRP-role “Manager (Auto)” by factory-default.
- Each MRA shall have assigned a priority (MRP_Prio) value. This MRP_Prio-value, combined with the MAC address, results in a finally unique priority. A lower MRP_Prio-value means a higher priority. If devices have identical MRP_Prio values, then the value of the device MAC address decides. A lower MAC address means a higher priority.
- At startup, each MRA network node shall start up with the manager voting process and shall send MRP_Test frames.
- Other MRA shall receive the MRP_Test frames and shall compare their own priority with the priority of the next MRA. If they do have the higher priority, they shall cause the remote manager that is sending the MRP_Test frame to withdraw from the MRA-role into the MRC-role by sending a negative test manager acknowledgement.
- The MRA or MRM with the highest priority shall become the one and only MRM of the ring and shall now start to manage the ring according to MRP.
- If the MRM, which has been elected via the manager voting option, is taken out of the ring or fails to operate, the voting process shall start again and the next priority MRA shall become the managing MRM.
- If an additional MRA is inserted into a ring containing an already elected MRM, then the already elected MRM and the newly inserted MRA will exchange MRP_Test frames and will thereby vote the new remaining MRM among each other.
- For the manager voting option to work correctly, all devices in the ring supporting the manager role shall be set to MRP-role “Manager (Auto)” or shall remain in the MRP-role “Client”. At least one device of a ring shall be set to either the MRP-role “Manager (Auto)” or “Manager” though.
- If an MRM without MRA capability is part of the network ring, regardless whether the ring is open or closed, all MRA shall be configured to the MRC role.

NOTE Configuring MRA to the MRC role is not performed automatically by the automanager voting process but is done by the user.

5.10.3 The manager voting process

The manager voting process shall be as follows:

- At power on, all MRA shall start with the manager voting process.
- Each MRA shall start to send MRP_Test frames on both ring-ports. The MRP_Test frame shall contain the MRAs priority value MRP_Prio.
- Received MRP_Test frames shall be forwarded to the local host only.
- The remote managers priority values contained in the received MRP_Test frames shall be compared with their own priority, as described in 5.10.2. If their own priority is higher than the received priority, a negative test manager acknowledgement (“MRP_TestMgrNAck”) frame, with the remote managers MAC address as additional information, shall be sent.
- If a negative test manager acknowledgement (“MRP_TestMgrNAck”) with their own MAC address is received, the receiving MRA shall initiate the transition into client (MRC) role by:
 - recording the MAC address and priority of the higher priority manager;
 - entering the client role and propagating the role change and the higher priority manager information via a “MRP_TestPropagate” frame sent on both ring-ports.
- On reaching the MRP client role it shall forward MRP_Test frames between ring ports like a standard MRC. Additionally, it shall also forward MRP_Test frames to the host CPU.

- From now on, the MRA in client role shall monitor whether the recorded higher priority manager is still alive, i. e. whether it still receives MRP_Test frames from that manager.
- The MRP_TestPropagate frame shall be used to inform other MRA devices in client role about the role change and the new higher priority manager. The clients receiving this frame shall update their higher priority manager information accordingly. This ensures that clients stay in client role in case of a role change of the monitored higher priority manager.
- The frames MRP_TestMgrNack and MRP_TestPropagate shall be sent using the destination multicast address MC_TEST.
- If the MRA in client role recognises (via MRP_Test frame reception monitoring timeout) that the higher priority manager has failed, it shall fall back into automanager role and shall thereby start a new manager voting process with the other MRA, which have fallen back into automanager role in the same way and for the same reason.

Figure 7 shows the frame exchanges over the time during the manager voting phase:

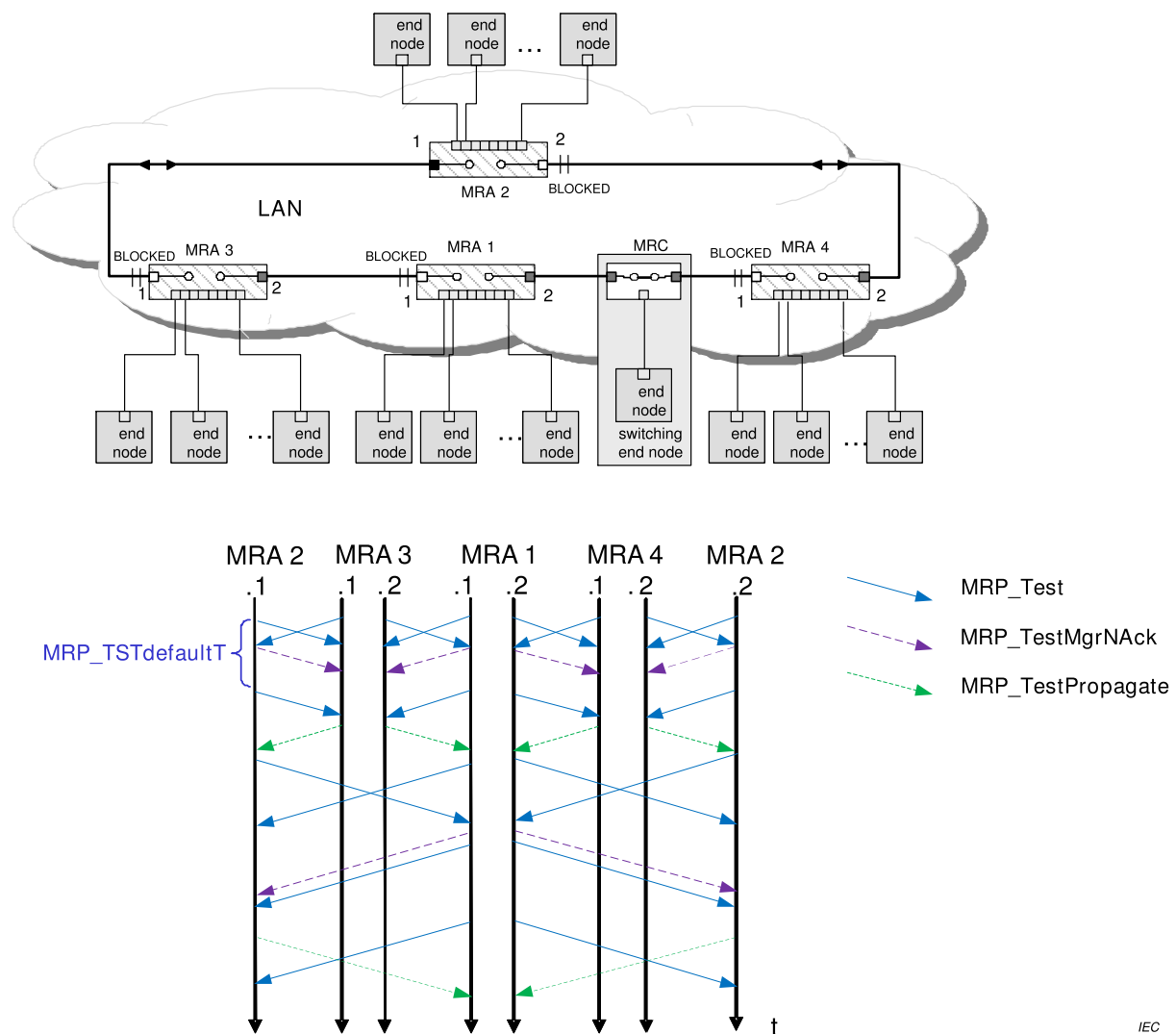


Figure 7 – Manager voting process

- A handshake (negative test acknowledgement) is required to ensure the higher priority manager is located in the ring. This is necessary, as due to the switch frame forwarding rules, which allow MRP_Test frames from a manager device located outside the ring, connected via its ring-port to a non-ring port of an MRP device being part of the ring, are also forwarded into the ring, if a switch without “inbound” filtering capability (“ingress filtering”) is used.

Figure 8 shows an MRP ring with one MRA (MRA2) located outside the ring. MRA2 is connected via one of its ring ports to a non-ring port of a device which is part of the ring. The frames follow the forwarding rules as defined in 5.2.

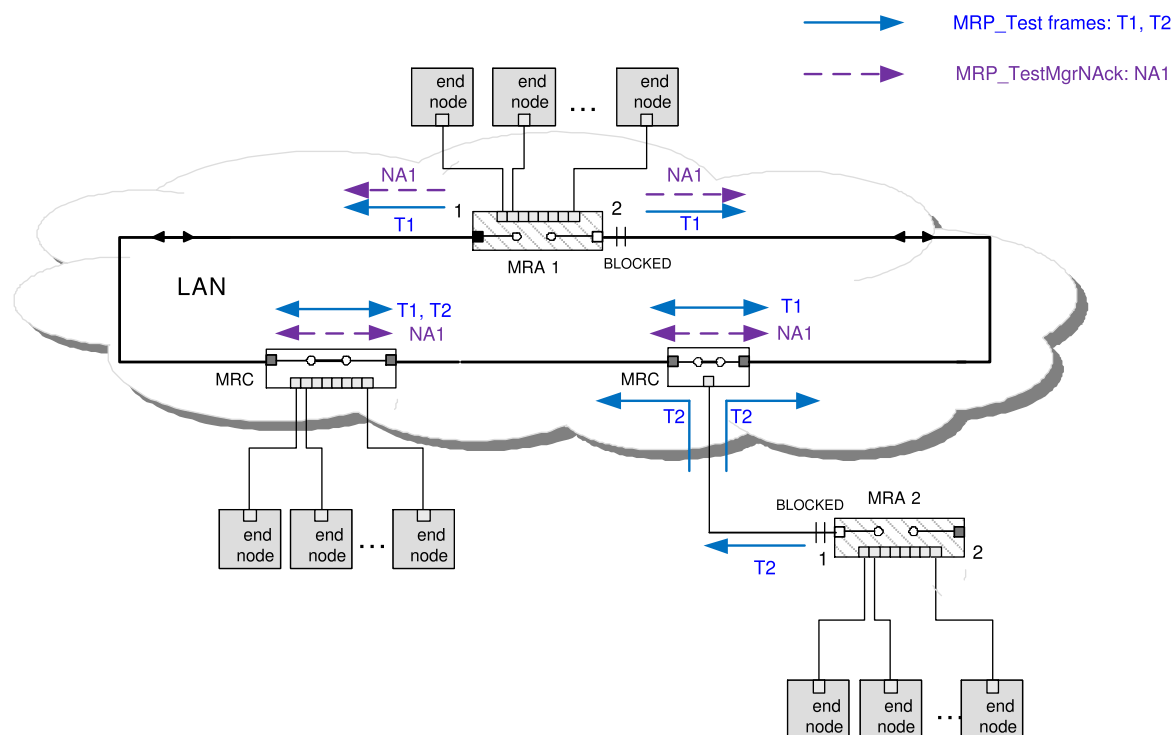


Figure 8 – MRA located outside the MRP ring

- While lower priority manager MRA1 receives test frames from higher priority manager MRA2, MRA2 does not receive test frames from lower priority manager MRA1 as these are not forwarded to non-ring ports, which excludes it from the voting process. Therefore MRA2 does not send negative test acknowledgements and MRA1 will stay in the MRM role. If MRA1 would have a higher priority as MRA2, MRA1 would send negative test acknowledgements but these would not reach MRA2 as these are not forwarded to non-ring ports either.

5.11 BLOCKED not supported (Option)

If an MRC is not able to support the BLOCKED port state at its ring ports, the MRC shall report it in the corresponding parameter of the MRP_LinkChange frames.

If an MRC does not support the BLOCKED state in a ring, then an MRM shall support additional functionalities (see Table 40 and Table 41, MRP_BLOCKED_SUPPORTED).

5.12 Interconnection port

The MIM and the MIC shall have one interconnection port.

The MIM and the MIC shall be able to detect the failure or recovery of a link on the interconnection port as follows:

- by using link detection mechanisms based on ISO/IEC/IEEE 8802-3 (IEEE 802.3)

In addition by applying one or both of the following mechanisms:

- a media link check protocol between interconnection ports. The media link check protocol shall be used at the MIM and MICs interconnection port if the MIM is configured to link check mode (LC-mode) and may be used if the MIM is configured to ring check mode (RC-mode).
- a periodic MRP_InTest frame which shall be sent and received by the MIM on its ring ports and interconnection port and shall be forwarded by the MICs on their ring ports and interconnection port. This mechanism shall be used for RC-mode and shall not be used for LC-mode.

The MIM and the MIC shall not forward MRP_InTest frames, MRP_InTopologyChange frames, MRP_InLinkStatusPoll frames and MRP_InLinkChange frames to ports other than ring ports and other than the interconnection port.

The interconnection port shall take one of the following port states:

- DISABLED
All frames shall be dropped.
- BLOCKED
All frames shall be dropped except the following:
 - MRP_InTest frames;
 - MRP_InLinkChange frames and MRP_InTopologyChange frames;
 - MRP_InLinkStatusPoll frames;
 - Media link check frames CFM-MRP according to IEEE 802.1Q-2011;
 - Frames specified in IEEE 802.1D-2004, Table 7-10 to pass ports in “Discarding” state (e.g. LLDP, IEEE 1588-2008 PTP peer to peer frames);
 - Frames only produced or consumed by the higher layer entities of this node and never forwarded.
- FORWARDING
All frames shall be passed through according to the forwarding behavior of IEEE 802.1D.

NOTE IEEE 802.1D refers to the port state corresponding to BLOCKED as “Discarding”.

5.13 Media redundancy Interconnection Manager (MIM)

The interconnection port of the MIM shall be connected to the interconnection port of an MIC in another MRP ring, and the interconnection port of an MIC in the same ring shall be connected to the interconnection port of an MIC in this other MRP ring, thereby forming an MRP interconnection topology as shown in Figure 9.

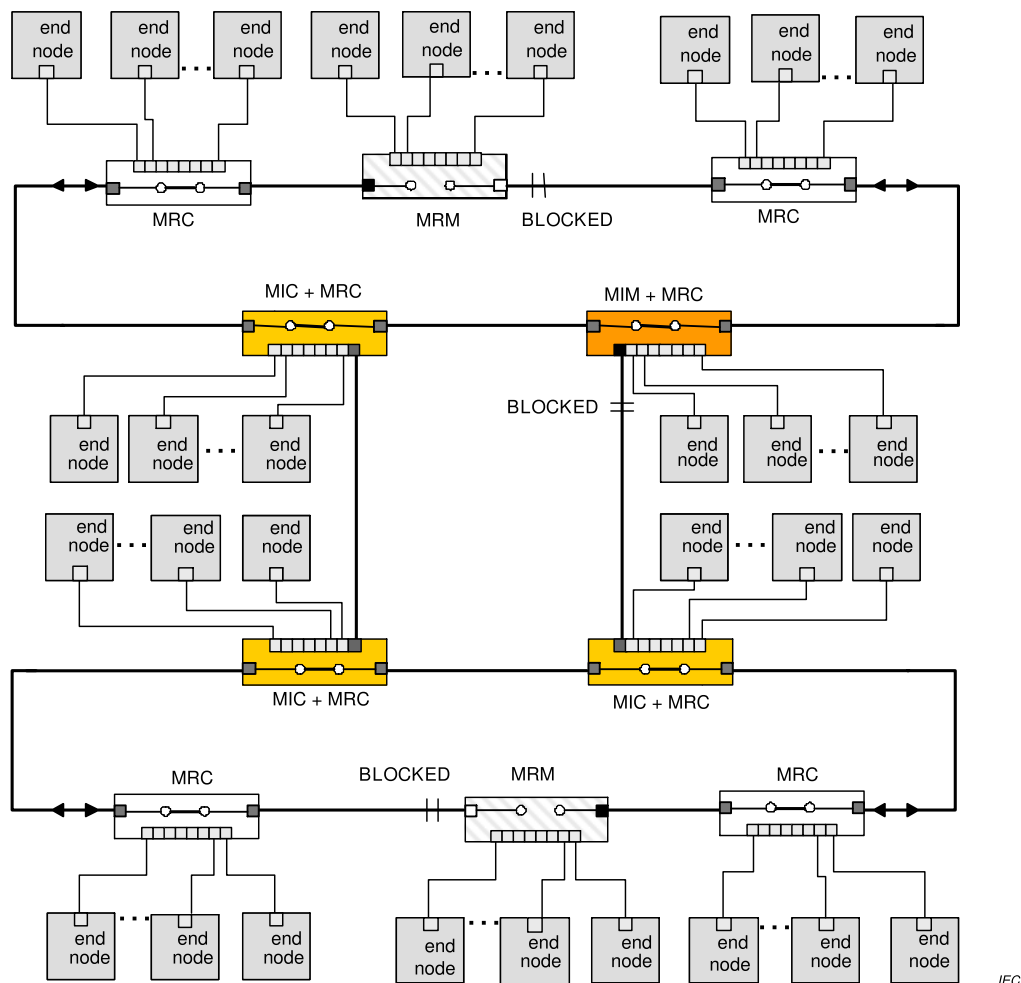


Figure 9 – MRP Interconnection topology

The MIM shall control its interconnection port state as follows:

- by directly reacting only on interconnection port link change notifications from the MICs and from its own interconnection port (link check mode: LC-mode);
- or optionally by reacting on MRP_InTest frames (ring check mode: RC-mode);
- setting its interconnection port in BLOCKED state if:
 - it is in LC-mode and receives interconnection port link up notifications from its own interconnection port and from the MICs of the interlink connection (this means that the interconnection topology is closed, see Figure 9);
 - it is in RC-mode and receives its own MRP_InTest frames (this means that the interconnection topology is closed, see Figure 9).
- setting the interconnection port in FORWARDING state if:
 - it is in LC-mode and receives an interconnection port link down notification from at least one of the MICs of the interlink connection (this means that the interconnection topology is open, see Figure 10.)
 - it is in RC-mode and does not receive its own MRP_InTest frames within a configured time according to MRP_IN_TSTdefaultT and MRP_IN_TSTNRmax in Table 59 (this means that the interconnection topology is open, see Figure 10).



The following mechanism supports synchronization between MIM and MIC at interconnection topology changes.

When it is in LC-mode, the MIM shall, after startup and after recognition of a link up at its interconnection port, issue a MRP_InLinkStatusPoll frame at its ring ports, to poll the status of the redundant interlink connection from the MICs.

When a transition to open or closed interconnection topology is detected, then the MIM shall send the `MRP_InTopologyChange` frames through both ring ports and through its interconnection port to the MICs and to the MRMs, to initiate the interconnection topology change.

If the MIC receives an MRP_InTopologyChange frame, then the MIC shall, after recognition of a link up at its interconnection port, change the port state of its interconnection port to FORWARDING.

If the MRMs receive a MRP_InTopologyChange frame, then the MRMs shall send MRP_TopologyChange frames at both ring ports, with the delay, after which all MRCs and MICs in the connected MRP rings will clear their Filtering Database (FDB).

The MIM shall not forward its own MRP_InTest frames between the ring ports and its interconnection port.

The MIM shall forward MRP_InLinkChange frames, MRP_InLinkStatusPoll frames and MRP_InTopologyChange frames received on one ring port to the other ring port and vice versa. The MIM shall process MRP_InLinkChange frames and MRP_InTopologyChange frames. The MIM shall not forward MRP_InLinkChange frames, MRP_InLinkStatusPoll frames and MRP_InTopologyChange frames, if the MIM received these frames at its interconnection port.

Each MIC shall send the configured delay in MRP_Interval to the MIM in the MRP_InLinkUp and MRP_InLinkDown frames to tell the MIM after which time the MIC will change its interconnection port state from BLOCKED to FORWARDING (MRP_InLinkUp frame) or from FORWARDING to BLOCKED (MRP_InLinkDown frame).

Each MIM shall support BLOCKED port state at the interconnection port.

5.14 Media redundancy Interconnection Client (MIC)

The interconnection port of the MIC shall be connected to the interconnection port of a MIM in another MRP ring, and the interconnection port of another MIC in the same ring shall be connected to the interconnection port of a MIC in this other MRP ring, thereby forming a MRP interconnection topology as shown in Figure 9.

Each MIC shall forward MRP_InTest frames received on one ring port to the other ring port and to the interconnection port. Each MIC shall forward MRP_InTest frames received on the interconnection port to both ring ports. If the MIC detects a failure or recovery of the interconnection port link, the MIC shall notify the change by sending MRP_InLinkChange frames through both of its ring ports. Each MIC shall forward MRP_InLinkChange frames received on one ring port to the other ring port and vice versa. Each MIC shall forward MRP_InLinkChange frames received on one of the ring ports to the interconnection port.

Each MIC shall forward MRP_InTopologyChange frames received on one ring port to the other ring port and vice versa. Each MIC shall process MRP_InTopologyChange frames. It shall, after recognition of a link up at its interconnection port, change the port state of its interconnection port to FORWARDING.

After receiving a MRP_InLinkStatusPoll frame from the MIM, each MIC shall respond with a MRP_InLinkChange status frame informing about its current interconnection port link status.

Each MIC shall support BLOCKED port state at the interconnection port.

5.15 Interconnection domain

The interconnection domain represents an interconnection topology. The MIM and the MICs of one interconnection topology belong to the same interconnection domain. A unique interconnection ID shall be allocated as a key attribute, especially if MIM and MICs are member of rings with multiple interconnections. A node shall assign exactly one unique interconnection port per interconnection domain.

Each interconnection topology in the network shall have a unique interconnection ID.

There shall be only one active MIM in an interconnection domain.

NOTE Interconnection ports are expected to behave as if STP, RSTP or MSTP is disabled.

5.16 Interconnection diagnosis

In an interconnection domain the following diagnosis events handling shall be implemented by each MIM.

- If a device is configured as MIM, but not operating in this role, it shall signal a “MRP_INTERCONNECTION_MANAGER_ROLE_FAIL” diagnosis event and suspend reporting of all other media redundancy interconnection protocol diagnosis events while not in the manager role.
- If a device is operating in MRP interconnection manager role (MIM) and detects or is informed about an open interconnection topology, it shall signal the “INTERCONNECTION_OPEN” event.

These events shall be signaled by using the State Change service (see 7.3).

6 MRP Class specification

6.1 General

The MRP Application Service Element (ASE) defines two object types.

6.2 Template

6.2.1 Media redundancy template

An MRP object is described by the following template:

ASE: Media redundancy ASE

CLASS: Media redundancy

CLASS ID: not used

PARENT CLASS: IEEE 802.3, IEEE 802.1D

ATTRIBUTES:

1.	(m)	Key Attribute:	Domain ID
2.	(m)	Attribute:	Domain Name
3.	(m)	Attribute:	Ring Port 1 ID
4.	(m)	Attribute:	Ring Port 2 ID
5.	(o)	Attribute:	VLAN ID
6.	(m)	Attribute:	Expected Role (MANAGER (AUTO), MANAGER, CLIENT)
7.	(c)	Constraint:	Expected Role = MANAGER
7.1	(m)	Attribute:	Manager Priority
7.2	(m)	Attribute:	Topology Change Interval
7.3	(m)	Attribute:	Topology Change Repeat Count
7.4	(m)	Attribute:	Short Test Interval
7.5	(m)	Attribute:	Default Test Interval
7.6	(m)	Attribute:	Test Monitoring Count
7.7	(m)	Attribute:	Non-blocking MRC supported (TRUE, FALSE)
7.8	(c)	Constraint:	Non-blocking MRC supported = TRUE
7.8.1	(m)	Attribute:	Test Monitoring Extended Count
7.9	(o)	Attribute:	React On Link Change (TRUE, FALSE)
7.10	(m)	Attribute:	Check Media Redundancy (TRUE, FALSE)
7.10.1	(c)	Constraint:	Check Media Redundancy = TRUE
7.10.1.1	(m)	Attribute:	Real Role State
7.10.1.2	(m)	Attribute:	Real Ring State
7.10.1.3	(o)	Attribute:	Ring Port 1 Port State
7.10.1.4	(o)	Attribute:	Ring Port 2 Port State
8.	(c)	Constraint:	Expected Role = CLIENT
8.1	(m)	Attribute:	Link Down Interval
8.2	(m)	Attribute:	Link Up Interval
8.3	(m)	Attribute:	Link Change Count
8.4	(o)	Attribute:	Ring Port 1 Port State
8.5	(o)	Attribute:	Ring Port 2 Port State
8.6	(m)	Attribute:	BLOCKED state supported (TRUE, FALSE)

SERVICES:

1	(m)	OpsService:	Start MRM
2	(m)	OpsService:	Stop MRM
3	(o)	OpsService:	State Change
4	(m)	OpsService:	Start MRC
5	(m)	OpsService:	Stop MRC
6	(o)	OpsService:	Read MRM
7	(o)	OpsService:	Read MRC

6.2.2 Media redundancy Interconnection template

An MRP Interconnection object is described by the following template:

ASE: Media redundancy interconnection ASE

CLASS: Media redundancy interconnection

CLASS ID: not used

PARENT CLASS: IEEE 802.3, IEEE 802.1D

ATTRIBUTES:

1.	(m)	Key Attribute:	Interconnection ID
2.	(m)	Attribute:	Interconnection Name
3.	(m)	Attribute:	Interconnection Port ID
4.	(o)	Attribute:	Interconnection Port State
5.	(o)	Attribute:	Interconnection VLAN ID
6.	(o)	Attribute:	Interconnection Mode
7.	(m)	Attribute:	Interconnection Role (MANAGER, CLIENT)
8.	(c)	Constraint:	Interconnection Role = MANAGER
8.1	(m)	Attribute:	Interconnection Topology Change Interval
8.2	(m)	Attribute:	Interconnection Topology Change Repeat Count
8.3	(o)	Attribute:	Interconnection Default Test Interval
8.4	(o)	Attribute:	Interconnection Test Monitoring Count
8.5	(o)	Attribute:	Interconnection Link Status Poll Interval
8.6	(o)	Attribute:	Interconnection Link Status Poll Repeat Count
8.7	(m)	Attribute:	Interconnection Topology State
9.	(c)	Constraint:	Interconnection Role = CLIENT
9.1	(m)	Attribute:	Interconnection Link Down Interval
9.2	(m)	Attribute:	Interconnection Link Up Interval
9.3	(m)	Attribute:	Interconnection Link Change Count

SERVICES:

1	(m)	OpsService:	Start MIM
2	(m)	OpsService:	Stop MIM
3	(o)	OpsService:	Interconnection State Change
4	(m)	OpsService:	Start MIC
5	(m)	OpsService:	Stop MIC
5	(o)	OpsService:	Read MIM
6	(o)	OpsService:	Read MIC

6.3 Attributes

Domain ID

This key attribute defines the redundancy domain representing the ring the MRP object belongs to. It is set to default Domain ID or provided as unique ID by engineering.

Attribute Type: UUID

Domain Name

This attribute defines the redundancy domain representing the ring the Media redundancy object belongs to. It is set to default Domain Name or provided as unique ID by engineering.

Attribute Type: VisibleString[240]

Ring Port 1 ID

This attribute specifies one port of a switch which is assigned as ring port 1 in the redundancy domain referenced by the value of the attribute Domain ID.

Attribute Type: Unsigned16

Ring Port 2 ID

This attribute specifies another port of a switch different from Ring Port 1 ID which is assigned as ring port 2 in the redundancy domain referenced by the value of the attribute Domain ID.

Attribute Type: Unsigned16

VLAN ID

This optional attribute may be used by the MRP object and specifies its VLAN identifier in the redundancy domain.

Attribute Type: Unsigned16

Expected Role

This attribute specifies the role of the MRP object in the redundancy domain.

Attribute Type: Unsigned16

Allowed values: MANAGER (Auto), MANAGER, CLIENT

Manager Priority

This attribute shall contain the priority of the MRM. A lower value indicates a higher priority, 0x0000 (highest priority) to 0xFFFF (lowest priority) in increments of 0x0001.

Attribute Type: Unsigned16

Topology Change Interval

This attribute specifies the interval for sending MRP_TopologyChange frames.

Attribute Type: Unsigned16

Topology Change Repeat Count

This attribute specifies the interval count which controls repeated transmissions of MRP_TopologyChange frames.

Attribute Type: Unsigned16

Short Test Interval

This attribute specifies the short interval for sending MRP_Test frames on ring ports after link changes in the ring.

Attribute Type: Unsigned16

Default Test Interval

This attribute specifies the default interval for sending MRP_Test frames on ring ports.

Attribute Type: Unsigned16

Test Monitoring Count

This attribute specifies the interval count for monitoring the reception of MRP_Test frames.

Attribute Type: Unsigned16

Non-blocking MRC supported

This attribute specifies the ability of the MRM to support MRCs without BLOCKED port state support in the ring.

Attribute Type: Boolean

Test Monitoring Extended Count

This attribute specifies the extended interval count for monitoring the reception of MRP_Test frames.

Attribute Type: Unsigned16

React On Link Change

This optional attribute specifies whether the MRM reacts immediately on MRP_LinkChange frames or not.

Attribute Type: Boolean

Check Media Redundancy

This attribute specifies whether monitoring of MRM state is enabled (TRUE) or disabled (FALSE) in the redundancy domain.

Attribute Type: Boolean

Real Role State

This attribute specifies the actual role of the MRP object in the redundancy domain.

Attribute Type: Unsigned16

Allowed values: MANAGER (Auto), MANAGER, CLIENT, UNDEFINED

Real Ring State

This attribute specifies the actual ring state of the MRP object in the redundancy domain. The Ring State shall have one of the following values:

OPEN: Ring is open due to link or MRC failure in ring.

CLOSED: Ring is closed (normal operation, no error).

UNDEFINED: Shall be set if the attribute Real Role State contains the value CLIENT (i.e. MRP object was reconfigured to client role).

Attribute Type: Unsigned16

Allowed values: OPEN, CLOSED, UNDEFINED

Ring Port 1 Port State

This optional attribute specifies the actual port state of Ring Port 1. The Ring Port 1 state shall be specified according to Ring port states in 5.2.

Attribute Type: Unsigned16

Allowed values: DISABLED, BLOCKED, FORWARDING

Ring Port 2 Port State

This optional attribute specifies the actual port state of Ring Port 2. The Ring Port 2 state shall be specified according to Ring port states in 5.2.

Attribute Type: Unsigned16

Allowed values: DISABLED, BLOCKED, FORWARDING

Link Down Interval

This attribute specifies the interval for sending MRP_LinkDown frames on ring ports.

Attribute Type: Unsigned16

Link Up Interval

This attribute specifies the interval for sending MRP_LinkUp frames on ring ports.

Attribute Type: Unsigned16

Link Change Count

This attribute specifies the MRP_LinkChange frame count which controls repeated transmission of MRP_LinkChange frames.

Attribute Type: Unsigned16

BLOCKED state supported

This attribute specifies whether the MRC supports BLOCKED state at its ring ports or not.

Attribute Type: Boolean

Interconnection ID

This key attribute defines the interconnection domain representing the interconnection topology the MRP interconnection object belongs to. It is provided as unique ID by engineering..

Attribute Type: Unsigned16

Interconnection Name

This attribute specifies the name of the interconnection domain referenced by the value of the attribute Interconnection ID. It is set to a default Interconnection Name or provided as Interconnection Name by engineering.

Attribute Type: VisibleString[240]

Interconnection Port ID

This attribute specifies one port of a switch which is assigned as interconnection port in the interconnection domain referenced by the value of the attribute Interconnection ID.

Attribute Type: Unsigned16

Interconnection Port State

This optional attribute specifies the actual port state of the interconnection port. The Interconnection Port state shall be specified according to Interconnection port states in 5.2.

Attribute Type: Unsigned16

Allowed values: DISABLED, BLOCKED, FORWARDING

Interconnection VLAN ID

This optional attribute may be used by the MRP interconnection object and specifies its VLAN identifier in the interconnection domain.

Attribute Type: Unsigned16

Interconnection Mode

This attribute specifies the mode of the MRP interconnection object in the interconnection domain.

LC_MODE: Link check mode. The MIM analyses the ring interconnection state by using the link change notifications of the MICs. MRP_InTest frames are disabled.

RC_MODE: Ring check mode. The MIM analyses the ring interconnection state by sending and receiving MRP_InTest frames. Link change notifications of the MICs may be used in addition to optimize the recovery timing.

Attribute Type: Unsigned16

Allowed values: UNDEFINED, LC_MODE, RC_MODE

Interconnection Role

This attribute specifies the role of the MRP interconnection object in the interconnection domain.

Attribute Type: Unsigned16

Allowed values: MANAGER, CLIENT

Interconnection Topology Change Interval

This attribute specifies the interval for sending MRP_InTopologyChange frames.

Attribute Type: Unsigned16

Interconnection Topology Change Repeat Count

This attribute specifies the interval count which controls repeated transmissions of MRP_InTopologyChange frames.

Attribute Type: Unsigned16

Interconnection Default Test Interval

This attribute specifies the default interval for sending MRP_InTest frames on ring ports and on the interconnection port.

Attribute Type: Unsigned16

Interconnection Test Monitoring Count

This attribute specifies the interval count for monitoring the reception of MRP_InTest frames.

Attribute Type: Unsigned16

Interconnection Link Status Poll Interval

This attribute specifies the interval for sending MRP_InLinkStatusPoll frames.

Attribute Type: Unsigned16

Interconnection Link Status Poll Repeat Count

This attribute specifies the interval count which controls repeated transmissions of MRP_InLinkStatusPoll frames.

Attribute Type: Unsigned16

Interconnection Topology State

This attribute specifies the actual interconnection topology state of the MRP interconnection object in the interconnection domain. The Interconnection Topology State shall have one of the following values:

UNDEFINED: Device has no information (yet) about the Interconnection topology;

OPEN: Interconnection topology is open due to link or MIC failure in the interconnection topology;

CLOSED: Interconnection topology is closed (normal operation, no error).

Attribute Type: Unsigned16

Allowed values: UNDEFINED, OPEN, CLOSED

Interconnection Link Down Interval

This attribute specifies the interval for sending MRP_InLinkDown frames on ring ports.

Attribute Type: Unsigned16

Interconnection Link Up Interval

This attribute specifies the interval for sending MRP_InLinkUp frames on ring ports.

Attribute Type: Unsigned16

Interconnection Link Change Count

This attribute specifies the MRP_InLinkChange frame count which controls repeated transmission of MRP_InLinkChange frames.

Attribute Type: Unsigned16

7 MRP Service specification

7.1 Start MRM

The Start MRM service creates a local instance of the MRM protocol machine.

Table 3 shows the parameters of the service.

Table 3 – MRP Start MRM

Parameter name	Req	Cnf
Argument	M	
Domain ID	M	
Ring Port 1 ID	M	
Ring Port 2 ID	M	
VLAN ID	U	
Manager Priority	U	
Topology Change Interval	U	
Topology Change Repeat Count	U	
Short Test Interval	U	
Default Test Interval	U	
Test Monitoring Count	U	
Non-blocking MRC supported	U	
Test Monitoring Extended Count	U	
React On Link Change	U	
Check Media Redundancy	U	
Result(+)		S
Domain ID		M
Result(-)		S
Domain ID		M
Error Code		M
The meaning of Req, Cnf, M, U and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Ring Port 1 ID

This parameter contains the ID of the port which serves as first ring port.

Ring Port 2 ID

This parameter contains the ID of the port which serves as second ring port.

VLAN ID

This optional parameter contains the value for the VLAN identifier.

Manager Priority

This parameter contains the value for the manager priority.

Topology Change Interval

This parameter contains the value of the interval for sending MRP_TopologyChange frames.

Topology Change Repeat Count

This parameter contains the value of the interval count which controls repeated transmissions of MRP_TopologyChange frames.

Short Test Interval

This parameter contains the value of the short interval for sending MRP_Test frames on ring ports after link changes in the ring.

Default Test Interval

This parameter contains the value of the default interval for sending MRP_Test frames on ring ports.

Test Monitoring Count

This parameter contains the value of the interval count for monitoring the reception of MRP_Test frames.

Non-blocking MRC supported

This parameter specifies the ability of the MRM to support MRCs without BLOCKED port state support in the ring.

Test Monitoring Extended Count

This optional parameter contains the value of the extended interval count for monitoring the reception of MRP_Test frames.

React On Link Change

This optional parameter specifies whether the MRM reacts on MRP_LinkChange frames or not.

Check Media Redundancy

This parameter selects whether monitoring of MRM state is enabled or disabled.

Result(+)

This parameter indicates that the service request succeeded.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: DOMAIN_ID_MISMATCH, ROLE_NOT_SUPPORTED, INVALID_RINGPORT

7.2 Stop MRM

This service shall be used to stop the MRM protocol machine. Ring port states and switch functionality remain.

Table 4 shows the parameters of the service.

Table 4 – MRP Stop MRM

Parameter name	Req	Cnf
Argument	M	
Domain ID	M	
Result(+)		S
Domain ID		M
Result(-)		S
Domain ID		M
Error Code		M
The meaning of Req, Cnf, M and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: DOMAIN_ID_MISMATCH

7.3 State Change

This service shall be used to indicate a change of the MRP domain state.

Table 5 shows the parameters of the service.

Table 5 – MRP Change State

Parameter name	Ind
Argument	M
Domain ID	M
Error Type List	M
The meaning of Ind and M is specified in ISO/IEC 10164-1.	

Argument

The argument shall convey the service specific parameters of the service request.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Error Type List

This attribute consists of the following elements:

Error Type

This attribute identifies a media redundancy error.

Attribute Type: Unsigned16

Allowed values: MANAGER_ROLE_FAIL, RING_OPEN, MULTIPLE_MANAGERS

Appear

This attribute identifies whether the error appears or disappears.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

7.4 Start MRC

The Start MRC service creates an instance of the MRC protocol machine.

Table 6 shows the parameters of the service.

Table 6 – MRP Start MRC

Parameter name	Req	Cnf
Argument	M	
Domain ID	M	
Ring Port 1 ID	M	
Ring Port 2 ID	M	
VLAN ID	U	
Link Down Interval	U	
Link Up Interval	U	
Link Change Count	U	
BLOCKED state supported	U	
Result(+)		S
Domain ID		M
Result(-)		S
Domain ID		M
Error Code		M
The meaning of Req, Cnf, M, U and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Ring Port 1 ID

This parameter contains the ID of the port which serves as first ring port.

Ring Port 2 ID

This parameter contains the ID of the port which serves as second ring port.

VLAN ID

This optional parameter contains the value for the VLAN identifier.

Link Down Interval

This parameter contains the value of the interval for sending MRP_LinkDown frames on ring ports.

Link Up Interval

This parameter contains the value of the interval for sending MRP_LinkUp frames on ring ports.

Link Change Count

This parameter contains the value of the MRP_LinkChange frame count which controls repeated transmissions of MRP_LinkUp or MRP_LinkDown frames.

BLOCKED state supported

This parameter specifies whether the MRC supports BLOCKED state at its ring ports or not.

Result(+)

This parameter indicates that the service request succeeded.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: DOMAIN_ID_MISMATCH, ROLE_NOT_SUPPORTED, INVALID_RINGPORT

7.5 Stop MRC

This service shall be used to stop the MRC protocol machine. Ring port states and switch functionality remain. Table 7 shows the parameters of the service.

Table 7 – MRP Stop MRC

Parameter name	Req	Cnf
Argument	M	
Domain ID	M	
Result(+)		S
Domain ID		M
Result(-)		S
Domain ID		M
Error Code		M
The meaning of Req, Cnf, M and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: DOMAIN_ID_MISMATCH

7.6 Read MRM

The optional Read MRM service reads the actual state of the MRM protocol machine.

Table 8 shows the parameters of the service.

Table 8 – MRP Read MRM

Parameter name	Req	Rsp
Argument	M	
Domain ID	M	
Result(+)		S
Domain ID		M
Ring Port 1 ID		M
Ring Port 2 ID		M
VLAN ID		U
Manager Priority		M
Check Media Redundancy		M
Real Role State		M
Real Ring State		M
Ring Port 1 Port State		U
Ring Port 2 Port State		U
Topology Change Interval		U
Topology Change Repeat Count		U
Short Test Interval		U
Default Test Interval		U
Test Monitoring Count		U
Non-blocking MRC supported		U
Test Monitoring Extended Count		U
React On Link Change		U
Result(-)		S
Domain ID		M
Error Code		M
The meaning of Req, Rsp, M, U and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Ring Port 1 ID

This parameter contains the ID of the port which serves as first ring port.

Ring Port 2 ID

This parameter contains the ID of the port which serves as second ring port.

VLAN ID

This optional parameter contains the value for the VLAN identifier.

Manager Priority

This parameter contains the value for the manager priority.

Check Media Redundancy

This parameter indicates whether monitoring of MRM is enabled or disabled

Topology Change Interval

This parameter contains the value of the interval for sending MRP_TopologyChange frames.

Topology Change Repeat Count

This parameter contains the value of the interval count which controls repeated transmission of MRP_TopologyChange frames.

Short Test Interval

This parameter contains the value of the short interval for sending MRP_Test frames on ring ports after link changes in the ring.

Default Test Interval

This parameter contains the value of the default interval for sending MRP_Test frames on ring ports.

Test Monitoring Count

This parameter contains the value of the interval count for monitoring the reception of MRP_Test frames.

Non-blocking MRC supported

This parameter contains the ability of the MRM to support MRC without BLOCKED port state support in the ring.

Test Monitoring Extended Count

This optional parameter contains the value of the extended interval count for monitoring the reception of MRP_Test frames.

Real Role State

This attribute contains the actual role of the MRP object in the redundancy domain.

Real Ring State

This attribute contains the actual ring state of the MRP object in the redundancy domain.

Ring Port 1 Port State

This optional attribute contains the actual port state of Ring Port 1.

Ring Port 2 Port State

This optional attribute contains the actual port state of Ring Port 2.

React On Link Change

This optional parameter contains whether the MRM reacts on MRP_LinkChange frames or not.

Result(-)

This parameter indicates that the service request failed.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: DOMAIN_ID_MISMATCH, MANAGER_READ_FAIL

7.7 Read MRC

The optional Read MRC service reads the actual state of the MRC protocol machine.

Table 9 shows the parameters of the service.

Table 9 – MRP Read MRC

Parameter name	Req	Rsp
Argument	M	
Domain ID	M	
Result(+)		S
Domain ID		M
Ring Port 1 ID		M
Ring Port 2 ID		M
VLAN ID		U
Ring Port 1 Port State		U
Ring Port 2 Port State		U
Link Down Interval		U
Link Up Interval		U
Link Change Count		U
BLOCKED state supported		U
Result(-)		S
Domain ID		M
Error Code		M
The meaning of Req, Rsp, M, U and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Ring Port 1 ID

This parameter contains the ID of the port which serves as first ring port.

Ring Port 2 ID

This parameter contains the ID of the port which serves as second ring port.

VLAN ID

This optional parameter contains the value for the VLAN identifier.

Link Down Interval

This parameter contains the value of the interval for sending MRP_LinkDown frames on ring ports.

Link Up Interval

This parameter contains the value of the interval for sending MRP_LinkUp frames on ring ports.

Link Change Count

This parameter contains the value of the MRP_LinkChange frame count which controls repeated transmission of MRP_LinkUp or MRP_LinkDown frames.

Ring Port 1 Port State

This optional attribute contains the actual port state of Ring Port 1.

Ring Port 2 Port State

This optional attribute contains the actual port state of Ring Port 2.

BLOCKED state supported

This parameter contains whether the MRC supports BLOCKED state at its ring ports or not.

Result(-)

This parameter indicates that the service request failed.

Domain ID

This is the key attribute to identify the instance of the protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: DOMAIN_ID_MISMATCH, CLIENT_READ_FAIL

7.8 Start MIM

The Start MIM service creates a local instance of the MIM protocol machine.

Table 10 shows the parameters of the service.

Table 10 – MRP Start MIM

Parameter name	Req	Cnf
Argument	M	
Interconnection ID	M	
Interconnection Name	M	
Interconnection Port ID	M	
Interconnection VLAN ID	U	
Interconnection Mode	M	
Interconnection Topology Change Interval	U	
Interconnection Topology Change Repeat Count	U	
Interconnection Default Test Interval	U	
Interconnection Test Monitoring Count	U	
Interconnection Link Status Poll Interval	U	
Interconnection Link Status Poll Repeat Count	U	
Result(+)		S
Interconnection ID		M
Result(-)		S
Interconnection ID		M
Error Code		M
The meaning of Req, Cnf, M, U and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Interconnection Name

This parameter contains the name of the interconnection domain.

Interconnection Port ID

This parameter contains the ID of the port which serves as interconnection port.

Interconnection VLAN ID

This optional parameter contains the value for the VLAN identifier.

Interconnection Mode

This parameter contains the requested operation mode of the MIM.

Interconnection Topology Change Interval

This parameter contains the value of the interval for sending MRP_InTopologyChange frames.

Interconnection Topology Change Repeat Count

This parameter contains the value of the interval count which controls repeated transmissions of MRP_InTopologyChange frames.

Interconnection Default Test Interval

This parameter contains the value of the default interval for sending MRP_InTest frames on ring ports and on the interconnection port.

Interconnection Test Monitoring Count

This parameter contains the value of the interval count for monitoring the reception of MRP_InTest frames.

Interconnection Link Status Poll Interval

This parameter contains the value of the interval for sending MRP_InLinkStatusPoll frames.

Interconnection Link Status Repeat Count

This parameter contains the value of the interval count which controls repeated transmissions of MRP_InLinkStatusPoll frames.

Result(+)

This parameter indicates that the service request succeeded.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Result(-)

This parameter indicates that the service request failed.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: INTERCONNECTION_ID_MISMATCH, ROLE_NOT_SUPPORTED, INVALID_INTERCONNECTIONPORT

7.9 Stop MIM

This service shall be used to stop the MIM protocol machine. Interconnection port state and switch functionality remain.

Table 11 shows the parameters of the service.

Table 11 – MRP Stop MIM

Parameter name	Req	Cnf
Argument	M	
Interconnection ID	M	
Result(+)		S
Interconnection ID		M
Result(-)		S
Interconnection ID		M
Error Code		M
The meaning of Req, Cnf, M and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Result(-)

This parameter indicates that the service request failed.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: INTERCONNECTION_ID_MISMATCH

7.10 Interconnection State Change

This optional service shall be used to indicate a change of the MRP interconnection domain state.

Table 12 shows the parameters of the service.

Table 12 – MRP Interconnection Change State

Parameter name	Ind
Argument	M
Interconnection ID	M
Error Type List	M
The meaning of Ind and M is specified in ISO/IEC 10164-1.	

Argument

The argument shall convey the service specific parameters of the service request.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Error Type List

This attribute consists of the following elements:

Error Type

This attribute identifies a media redundancy interconnection error.

Attribute Type: Unsigned16

Allowed values: INTERCONNECTION_OPEN

Appear

This attribute identifies whether the error appears or disappears.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

7.11 Start MIC

The Start MIC service creates an instance of the MIC protocol machine.

Table 13 shows the parameters of the service.

Table 13 – MRP Start MIC

Parameter name	Req	Cnf
Argument	M	
Interconnection ID	M	
Interconnection Name	M	
Interconnection Port ID	M	
Interconnection VLAN ID	U	
Interconnection Mode	M	
Interconnection Link Down Interval	U	
Interconnection Link Up Interval	U	
Interconnection Link Change Count	U	
Result(+)		S
Interconnection ID		M
Result(-)		S
Interconnection ID		M
Error Code		M
The meaning of Req, Cnf, M, U and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Interconnection Name

This parameter contains the name of the interconnection domain.

Interconnection Port ID

This parameter contains the ID of the port which serves as interconnection port.

Interconnection VLAN ID

This optional parameter contains the value for the VLAN identifier.

Interconnection Mode

This parameter contains the requested operation mode of the MIC.

Interconnection Link Down Interval

This parameter contains the value of the interval for sending MRP_InLinkDown frames on ring ports.

Interconnection Link Up Interval

This parameter contains the value of the interval for sending MRP_InLinkUp frames on ring ports.

Interconnection Link Change Count

This parameter contains the value of the MRP_InLinkChange frame count which controls repeated transmissions of MRP_InLinkUp or MRP_InLinkDown frames.

Result(+)

This parameter indicates that the service request succeeded.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Result(-)

This parameter indicates that the service request failed.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: INTERCONNECTION_ID_MISMATCH, ROLE_NOT_SUPPORTED, INVALID_INTERCONNECTIONPORT

7.12 Stop MIC

This service shall be used to stop the MIC protocol machine. Interconnection port state and switch functionality remain.

Table 14 shows the parameters of the service.

Table 14 – MRP Stop MIC

Parameter name	Req	Cnf
Argument	M	
Interconnection ID	M	
Result(+)		S
Interconnection ID		M
Result(-)		S
Interconnection ID		M
Error Code		M
The meaning of Req, Cnf, M and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Result(-)

This parameter indicates that the service request failed.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: INTERCONNECTION_ID_MISMATCH

7.13 Read MIM

The optional Read MIM service reads the actual state of the MIM protocol machine.

Table 15 shows the parameters of the service.

Table 15 – MRP Read MIM

Parameter name	Req	Rsp
Argument	M	
Interconnection ID	M	
Result(+)		S
Interconnection ID		M
Interconnection Name		M
Interconnection Port ID		M
Interconnection Port State		U
Interconnection VLAN ID		U
Interconnection Mode		M
Interconnection Topology Change Interval		U
Interconnection Topology Change Repeat Count		U
Interconnection Default Test Interval		U
Interconnection Test Monitoring Count		U
Interconnection Link Status Poll Interval		U
Interconnection Link Status Poll Repeat Count		U
Interconnection Topology State		M
Result(-)		S
Interconnection ID		M
Error Code		M
The meaning of Req, Rsp, M, U and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Interconnection Name

This parameter contains the name of the interconnection domain.

Interconnection Port ID

This parameter contains the ID of the port which serves as interconnection port.

Interconnection Port State

This optional attribute contains the actual port state of the interconnection port.

Interconnection VLAN ID

This optional parameter contains the value for the VLAN identifier.

Interconnection Mode

This parameter contains the operation mode of the MIM.

Interconnection Topology Change Interval

This parameter contains the value of the interval for sending MRP_InTopologyChange frames.

Interconnection Topology Change Repeat Count

This parameter contains the value of the interval count which controls repeated transmission of MRP_InTopologyChange frames.

Interconnection Default Test Interval

This parameter contains the value of the default interval for sending MRP_InTest frames on ring ports.

Interconnection Test Monitoring Count

This parameter contains the value of the interval count for monitoring the reception of MRP_InTest frames.

Interconnection Link Status Poll Interval

This parameter contains the value of the interval for sending MRP_InLinkStatusPoll frames.

Interconnection Link Status Poll Repeat Count

This parameter contains the value of the interval count for repeated transmissions of MRP_InLinkStatusPoll frames.

Interconnection Topology State

This attribute contains the actual interconnection topology state of the MRP interconnection object in the interconnection domain.

Result(-)

This parameter indicates that the service request failed.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: INTERCONNECTION_ID_MISMATCH, MANAGER_READ_FAIL

7.14 Read MIC

The optional Read MIC service reads the actual state of the MIC protocol machine.

Table 16 shows the parameters of the service.

Table 16 – MRP Read MIC

Parameter name	Req	Rsp
Argument	M	
Interconnection ID	M	
Result(+)		S
Interconnection ID		M
Interconnection Name		M
Interconnection Port ID		M
Interconnection Port State		U
Interconnection VLAN ID		U
Interconnection Mode		M
Interconnection Link Down Interval		U
Interconnection Link Up Interval		U
Interconnection Link Change Count		U
Result(-)		S
Interconnection ID		M
Error Code		M
The meaning of Req, Rsp, M, U and S is specified in ISO/IEC 10164-1.		

Argument

The argument shall convey the service specific parameters of the service request.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Interconnection Name

This parameter contains the name of the interconnection domain.

Interconnection Port ID

This parameter contains the ID of the port which serves as interconnection port.

Interconnection Port State

This optional attribute contains the actual port state of the interconnection port.

Interconnection VLAN ID

This optional parameter contains the value for the VLAN identifier.

Interconnection Mode

This parameter contains the operation mode of the MIC.

Interconnection Link Down Interval

This parameter contains the value of the interval for sending MRP_InLinkDown frames on ring ports.

Interconnection Link Up Interval

This parameter contains the value of the interval for sending MRP_InLinkUp frames on ring ports.

Interconnection Link Change Count

This parameter contains the value of the MRP_InLinkChange frame count which controls repeated transmission of MRP_InLinkUp or MRP_InLinkDown frames.

Result(-)

This parameter indicates that the service request failed.

Interconnection ID

This is the key attribute to identify the instance of the interconnection protocol machine.

Error Code

The parameter Result contains the error code of the specific error.

Type: Unsigned16

Allowed values: INTERCONNECTION_ID_MISMATCH, CLIENT_READ_FAIL

8 MRP protocol specification

8.1 PDU description

8.1.1 Basic data types

The conventions for this specification are according to IEC 61158-6-10:2014, 4.2. Notation and encoding of basic data types are according to IEC 61158-6-10:2014, 4.1.2 and IEC 61158-6-10:2014, 4.2.

8.1.2 DLPDU abstract syntax reference

Transfer syntax and encoding of the MRP protocol specification is according to IEC 61158-6-10:2014, 4.2.

The encoding and decoding of the fields in Table 17 shall be according to ISO/IEC/IEEE 8802-3 (IEEE 802.3) for the DLPDU.

Table 17 – MRP DLPDU syntax for ISO/IEC/IEEE 8802-3 (IEEE 802.3)

DLPDU name	DLPDU structure
DLPDU	Preamble ^a , StartFrameDelimiter, DestinationAddress, SourceAddress, DLSDU ^b , DLPDU_Padding ^c , FrameCheckSequence
DLSDU	[VLAN] ^d , LT, MRP-PDU
VLAN	LT(=0x8100), TagControlInformation
NOTE 1 According to ISO/IEC/IEEE 8802-3 (IEEE 802.3), the DLPDUs have a minimum length of 64 octets (excluding Preamble and StartFrameDelimiter).	
NOTE 2 For ISO/IEC/IEEE 8802-3 (IEEE 802.3), frames with VLAN tag the minimum frame size is increased to 68 octets in order to guarantee the minimum frame size of 64 octets after removing the VLAN tag by a bridge.	
^a The field contains at least 7 octets ^b The minimum DLSDU size is 2 octets. ^c The number of padding octets shall be in the range of 0 to 46 depending on the DLSDU size. The value shall be set to zero. ^d The VLAN field can be omitted in case of optimized transportation. The field VLAN may be set by the encoder but it may be discarded by intermediate bridges. The decoder shall accept DLPDUs with or without VLAN fields.	

8.1.3 Coding of the DLPDU field SourceAddress

This field shall be coded as data type octetString[6]. The value of the field SourceAddress shall be according to IEEE 802 MAC address, see IEEE 802-1D:2004, Clause 7. The port

MAC address is used for MRP DLPDU. The interface MAC address shall be different from any port MAC address.

8.1.4 Coding of the DLPDU field DestinationAddress

This field shall be coded as data type octetString[6].

The IEEE Organizationally Unique Identifier for MRP is 00-15-4E. It shall be set according to Table 18.

Table 18 – MRP OUI

Value for OUI (hexadecimal)	Meaning
00-15-4E	Global administered individual unicast
01-15-4E	Global administered group (multicast) address
02-15-4E	Local administered individual unicast
03-15-4E	Local administered group (multicast) address

For MRP-PDUs, the destination address value shall be set according to Table 19.

Table 19 – MRP MulticastMACAddress

Value OUI (Multicast) (hexadecimal)	Value ExtensionIdentifier (hexadecimal)	Meaning
01-15-4E	00-00-00	Reserved
01-15-4E	00-00-01	MC_TEST, used for MRP_Test frames
01-15-4E	00-00-02	MC_CONTROL, used for MRP_LinkChange, MRP_TopologyChange and MRP_Option frames
01-15-4E	00-00-03	MC_INTEST, used for MRP_InTest frames
01-15-4E	00-00-04	MC_INCONTROL, used for MRP_InLinkChange, MRP_InTopologyChange and MRP_InLinkStatusPoll frames
01-15-4E	00-00-05 to FF-FF-FF	Reserved

NOTE 1 Octet 1 contains the Individual/Group Address Bit (LSB).

NOTE 2 The destination multicast address MC_TEST is also used for MRP_TestMgrNAck and MRP_TestPropagate frames.

8.1.5 Coding of the field TagControlInformation

This field shall be coded according to IEEE 802.1Q as data type Unsigned16. The individual bits shall have the following meaning:

Bit 0 – 11: TagControlInformation.VLAN_Identifier
These bits shall be coded according to IEEE 802.1Q.

NOTE VLAN_Identifier 0 means that no VLANs are used.

Bit 12: TagControlInformation.CanonicalFormatIdentifier
These bits shall be coded according to IEEE 802.1Q.

NOTE CFI is constant 0.

Bit 13 – 15: TagControlInformation.Priority
These bits shall be coded according to IEEE 802.1Q.

For MRP, use of the VLAN field is optional. If present, the value of TagControlInformation.Priority shall be set according to Table 20.

Table 20 – MRP TagControlInformation.Priority field

Value (hexadecimal)	Meaning
0x07	MRP-PDU

8.1.6 Coding of the field LT

This field shall be coded as data type Unsigned16 with the values according to ISO/IEC/IEEE 8802-3 (IEEE 802.3). For MRP the value shall be set according to Table 21.

Table 21 – MRP LT field

Value (hexadecimal)	Meaning
0x88E3	MRP-PDU

8.1.7 MRP APDU abstract syntax

Table 22 defines the abstract syntax of the MRP-PDUs referred to as APDUs. The defined order of octets shall be used to convey the APDUs.

Table 22 – MRP APDU syntax

APDU name	APDU structure
MRP-PDU	MRP_Version, MRP_Type, MRP_Common, [MRP_Option], MRP_End, [Padding*] ^a
^a If the frame is shorter than 64 octets, it shall be extended with padding to 64 octets, according to ISO/IEC/IEEE 8802-3 (IEEE 802.3).	

Table 23 defines structures for substitutions of elements of the APDU structure shown in Table 22.

Table 23 – MRP Substitutions

Substitution name	Structure
MRP_Type	MRP_Test ^ MRP_LinkChange ^ MRP_TopologyChange ^ MRP_Option ^ MRP_InTest ^ MRP_InLinkChange ^ MRP_InLinkStatusPoll ^ MRP_InTopologyChange
MRP_Common	MRP_TLVHeader, MRP_SequenceID, MRP_DomainUUID
MRP_Option ^c	MRP_TLVHeader, MRP_OUI, MRP_SubOption1, [MRP_SubOption2*] , [Padding*] ^a
MRP_OUI	MRP_ManufacturerOUI ^ MRP_IECOUI
MRP_SubOption1	MRP_Ed1Type, MRP_Ed1ManufacturerData
MRP_SubOption2 ^d	MRP_TestMgrNack ^ MRP_TestPropagate ^ MRP_AutoMgr ^ MRP_ManufacturerFkt
MRP_TestMgrNack ^e	MRP_SubTLVHeader, MRP_Prio, MRP_SA, MRP_OtherMRMPrio, MRP_OtherMRMSA, [Padding*] ^a
MRP_TestPropagate ^e	MRP_SubTLVHeader, MRP_Prio, MRP_SA, MRP_OtherMRMPrio, MRP_OtherMRMSA, [Padding*] ^a
MRP_AutoMgr ^f	MRP_SubTLVHeader (=0x0300)
MRP_ManufacturerFkt	MRP_SubTLVHeader, MRP_ManufacturerData
MRP_End	MRP_TLVHeader (=0x0000)
MRP_Test ^b	MRP_TLVHeader, MRP_Prio, MRP_SA, MRP_PortRole, MRP_RingState, MRP_Transition, MRP_TimeStamp, [Padding*] ^a
MRP_TopologyChange	MRP_TLVHeader, MRP_Prio, MRP_SA, MRP_Interval, [Padding*] ^a
MRP_LinkChange	MRP_LinkUp ^ MRP_LinkDown
MRP_LinkDown	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_Interval, MRP_Blocked, [Padding*] ^a
MRP_LinkUp	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_Interval, MRP_Blocked, [Padding*] ^a
MRP_InTest	MRP_TLVHeader, MRP_InID, MRP_SA, MRP_PortRole, MRP_InState, MRP_Transition, MRP_TimeStamp, [Padding*] ^a
MRP_InLinkChange	MRP_InLinkUp ^ MRP_InLinkDown
MRP_InLinkDown	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_InID, MRP_Interval, [Padding*] ^a
MRP_InLinkUp	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_InID, MRP_Interval, [Padding*] ^a
MRP_InLinkStatusPoll	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_InID, [Padding*] ^a
MRP_InTopologyChange	MRP_TLVHeader, MRP_SA, MRP_InID, MRP_Interval, [Padding*] ^a
^a 32 bit alignment shall be ensured.	
^b In connection with high speed automation systems the length of the MRP_PDU of type MRP_Test is critical and should be kept as small as possible. Consult the appropriate automation system standards for frame length limitations.	
^c Frames of MRP_Type = MRP_Option shall not contain more than one MRP_SubOption2 with MRP_SubTLVHeader.Type between 0x01 and 0xf0.	
^d Frames of MRP_Type != MRP_Option which contain multiple MRP_SubOption2 shall arrange MRP_SubTLVHeader.Type in ascending order.	
^e Only to be used in MRP_Type = MRP_Option.	
^f Only to be used in MRP_Type = MRP_Test. Additional MRP_SubOption2 may only be manufacturer specific with MRP_SubTLVHeader.Type between 0xf1 and 0xff.	

8.1.8 Coding of the field MRP_TLVHeader

The coding of this field shall be according to IEC 61158-6-10:2014, 4.2.1 and the individual bits shall have the following meaning:

Bit 0 – 7: MRP_TLVHeader.Length

The value contains the number of subsequent octets of the according block.

Bit 8 – 15: MRP_TLVHeader.Type

This field shall be coded with the values according to Table 24.

Table 24 – MRP_TLVHeader.Type

Value (hexadecimal)	Meaning	Usage
0x00	MRP_End (MRP_TLVHeader.Length shall be set to zero)	Mandatory
0x01	MRP_Common	Mandatory
0x02	MRP_Test	Mandatory
0x03	MRP_TopologyChange	Mandatory
0x04	MRP_LinkDown	Mandatory
0x05	MRP_LinkUp	Mandatory
0x06	MRP_InTest	Mandatory in RC-mode
0x07	MRP_InTopologyChange	Mandatory
0x08	MRP_InLinkDown	Mandatory
0x09	MRP_InLinkUp	Mandatory
0x0A	MRP_InLinkStatusPoll	Mandatory in LC-mode
0x0B – 0x7E	Reserved	—
0x7F	MRP_Option (Organizationally specific)	Optional
0x80-0xFF	Reserved	—

8.1.9 Coding of the field MRP_SubTLVHeader

The coding of this field shall be according to IEC 61158-6-10:2014, 4.2.1 and the individual bits shall have the following meaning:

Bit 0 – 7: MRP_SubTLVHeader.Length

The value contains the number of subsequent octets of the according block.

Bit 8 – 15: MRP_SubTLVHeader.Type

This field shall be coded with the values according to Table 25.

Table 25 – MRP_SubTLVHeader.Type

Value (hexadecimal)	OUI	Meaning	Usage
0x00	don't care	Reserved	Mandatory
0x01	ManufacturerOUI or IECOUI	MRP_TestMgrNAck	Mandatory
0x02	ManufacturerOUI or IECOUI	MRP_TestPropagate	Mandatory
0x03	ManufacturerOUI or IECOUI	MRP_AutoMgr	Mandatory
0x04 – 0xF0	ManufacturerOUI or IECOUI	Reserved for IEC specific functions	Optional
0xF1 – 0xFF	ManufacturerOUI	Manufacturer specific functions	Optional

8.1.10 Coding of the field MRP_Ed1Type and MRP_Ed1ManufacturerData

The field MRP_Ed1Type shall be coded as data type Unsigned8.

For MRP_OUI = MRP_ManufacturerOUI the fields content shall be coded with the values according to Table 26.

Table 26 – MRP_Ed1Type and MRP_Ed1ManufacturerData

MRP_Ed1Type Value (hexadecimal)	MRP_Ed1ManufacturerData Content	MRP_Ed1ManufacturerData Length (Byte)	Usage
0x00	don't care	2	Mandatory
0x01 – 0x03	-	0	Mandatory
0x04	don't care	26	Mandatory
0x05 – 0xFE	-	0	Mandatory
0xFF	-	0	Mandatory

For MRP_OUI = IEC OUI the fields content shall be coded with the values according to Table 27.

Table 27 – MRP_Ed1Type and MRP_Ed1ManufacturerData

MRP_Ed1Type Value (hexadecimal)	MRP_Ed1ManufacturerData Content	MRP_Ed1ManufacturerData Length (Byte)	Usage
0x00 – 0xFE	Reserved	0	Mandatory
0xFF	don't care	0	Mandatory

8.1.11 Coding of the field MRP_Version

This field shall be coded as data type Unsigned16 with the values according to Table 28.

Table 28 – MRP_Version

Value (decimal)	Meaning
0	Reserved
1	Initial version of MRP
2 ... 65 535	Reserved

8.1.12 Coding of the field MRP_SequenceID

This field shall be coded as data type Unsigned16. It is used to identify the duplication of MRP frames in the ring. The range is from 0 to 65 535. The requesting application process shall provide a unique sequence number to each outstanding service request.

8.1.13 Coding of the field MRP_SA

This field shall be coded as data type octetString[6]. The value of the field MRP_SA shall be according to IEEE 802 MAC address and shall contain the MAC address of the sending switch host (Interface MAC Address).

8.1.14 Coding of the field MRP_OtherMRMSA

This field shall be coded as data type octetString[6]. The value of the field MRP_OtherMRMSA shall be according to IEEE 802 MAC address and shall be set according to Table 29.

Table 29 – Coding of the field MRP_OtherMRMSA

MRP_SubOption2	Value of MRP_OtherMRMSA
MRP_TestMgrNAck	Interface-MAC address of the lower priority manager whose test frame was received previously.
MRP_TestPropagate	Interface-MAC address of the higher priority manager whose MRP_TestMgrNAck frame was received previously.

8.1.15 Coding of the field MRP_Prio

This field shall be coded as data type Unsigned16 and set according to Table 30.

Table 30 – MRP_Prio

Value (hexadecimal)	Meaning
0x0000	MRM highest priority
0x1000 – 0x7000	MRM high priorities
0x8000	MRM default priority
0x9000 – 0x9FFF	MRA high priorities
0xA000	MRA default priority
0xA001-0xF000	MRA low priorities
0xFFFF	MRA lowest priority
Other	Reserved

NOTE Having one or more devices that are configured to MRA role and a manually configured MRM in the same ring is not supported in this edition of IEC 62439-2.

8.1.16 Coding of the field MRP_OtherMRMPrio

This field shall be coded as data type Unsigned16. The value of the field MRP_OtherMRMPrio shall be set according to Table 31

Table 31 – Coding of the field MRP_OtherMRMPrio

MRP_SubOption2	Value of MRP_OtherMRMPrio
MRP_TestMgrNAck	The field is left empty and will not be analysed by the receiver.
MRP_TestPropagate	Priority of the higher priority manager whose TestMgrNAck frame was received previously

8.1.17 Coding of the field MRP_PortRole

This field shall be coded as data type Unsigned16. The coding shall be according to Table 32.

Table 32 – MRP_PortRole

Value (hexadecimal)	Meaning	Usage
0x0000	Primary ring port	Frame is sent on primary ring port
0x0001	Secondary ring port	Frame is sent on secondary ring port
0x0002	Interconnection port	Frame is sent on interconnection port
0x0003 – 0xFFFF	Reserved	

8.1.18 Coding of the field MRP_RingState

This field shall be coded as data type Unsigned16 with the values according to Table 33.

Table 33 – MRP_RingState

Value (hexadecimal)	Meaning	Usage
0x0000	Ring open	MRM in Ring open state
0x0001	Ring closed	MRM in Ring closed state
0x0002 – 0xFFFF	Reserved	—

8.1.19 Coding of the field MRP_Interval

This field shall be coded as data type Unsigned16 with the values according to Table 34.

Table 34 – MRP_Interval

Value (hexadecimal)	Meaning	Usage
0x0000 – 0x07D0	Interval for next topology change event (in ms)	Mandatory
0x07D1 – 0xFFFF	Interval for next topology change event (in ms)	Optional

8.1.20 Coding of the field MRP_Transition

This field shall be coded as data type Unsigned16 with the values according to Table 35.

Table 35 – MRP_Transition

Value (hexadecimal)	Meaning	Usage
0x0000 – 0xFFFF	Number of transitions between ring open state and ring closed state, or number of transitions between interconnection topology open state and interconnection topology closed state.	Used for monitoring this value via a packet sniffer station

8.1.21 Coding of the field MRP_TimeStamp

This field shall be coded as data type Unsigned32 with the values according to Table 36.

Table 36 – MRP_TimeStamp

Value (hexadecimal)	Meaning	Usage
0x00000000 – 0xFFFFFFFF	Actual local counter value of 1ms counter	The value is used by the MRM to determine the maximum travel time of the MRP_Test frames in a ring, or the value is used by the MIM to determine the maximum travel time of the MRP_InTest frames in an interconnection topology.

8.1.22 Coding of the field MRP_Blocked

This field shall be coded as data type Unsigned16 with the values according to Table 37.

Table 37 – MRP_Blocked

Value (decimal)	Meaning	Usage
0	The MRC is not able to receive and forward MRP_Test frames, MRP_LinkChange frames and MRP_TopologyChange frames at a ring port whose port state is BLOCKED	Optional
1	The MRC is able to receive and forward MRP_Test frames, MRP_LinkChange frames and MRP_TopologyChange frames at a ring port whose port state is BLOCKED	Mandatory
2 ... 65 535	Reserved	—

8.1.23 Coding of the field MRP_ManufacturerOUI

This field shall be coded as data type `octetString[3]` with the Organizationally Unique Identifier (OUI) of the device manufacturer as defined by the IEEE Registration Authority Committee (RAC).

8.1.24 Coding of the field MRP_IECOUI

This field shall be coded as data type `octetString[3]` with the Organizationally Unique Identifier (OUI) of the IEC Organisation which is 00-15-4e as defined by the IEEE Registration Authority Committee (RAC).

8.1.25 Coding of the field MRP_ManufacturerData

This field shall be reserved for vendor specific data.

8.1.26 Coding of the field MRP_DomainUUID

This field shall be coded as UUID with the values according to Table 38.

Table 38 – MRP_DomainUUID

Value (hexadecimal)	Meaning	Usage
0x00000000-0000-0000-0000-000000000000		Reserved
0x00000000-0000-0000-0000-000000000001 – 0xFFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFFFFE	UUID for MRP redundancy domain	Optional
0xFFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFFFFF	Default UUID for MRP redundancy domain	Mandatory

8.1.27 Coding of the field MRP_InState

This field shall be coded as data type `Unsigned16` with the values according to Table 39.

Table 39 – MRP_InState

Value (hexadecimal)	Meaning	Usage
0x0000	Interconnection open	MIM in Interconnection topology open state
0x0001	Interconnection closed	MIM in Interconnection topology closed state
0x0002 – 0xFFFF	Reserved	—

8.1.28 Coding of the field MRP_InID

This field shall be coded as data type Unsigned16. It is used to identify the interconnection domain. The range is from 0 to 65 535. The requesting application process shall provide a unique MRP_InID to each interconnection topology.

8.2 Protocol machines

8.2.1 MRM protocol machine

The MRM protocol machine is defined in Table 41. The principal behavior of the protocol machine is shown in Figure 11.

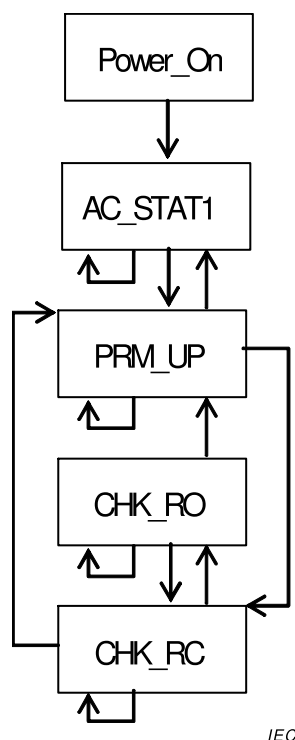


Figure 11 – MRP protocol machine for MRM

The text below is an explanation of the overall actions performed in the states. If a difference in the interpretation occurs between this text and the state machine, then the state machine supersedes.

Power_On

Initialization, the MRM shall start with both ring ports RPort_1 and RPort_2 in the port state BLOCKED. Static FDB entries for MRP multicast addresses MC_TEST, MC_CONTROL and MC_INCONTROL to host are generated. All MRP-PDU shall use the highest priority (ORG).

AC_STAT1 (Awaiting Connection State 1)

Startup, waiting for the first Link Up at one of its ring ports (called primary ring port), starting test monitoring of the ring, and transition to PRM_UP.

PRM_UP (Primary Ring Port with Link Up)

This state shall be reached if only the primary ring port has a link (secondary ring port with no link). The MRM shall send MRP_Test frames periodically through both ring ports even if the other ring port (secondary ring port) detected no link.

CHK_RO (Check Ring, Ring Open State)

The MRM did not receive its MRP_Test frames for a determined time, the MRP_RingState shall be set to ring open state.

This state can also be entered on reception of an MRP_LinkDown frame if the option MRP_REACT_ON_LINK_CHANGE is supported.

CHK_RC (Check Ring, Ring Closed State)

The MRM shall send its MRP_Test frames and shall check the link of its ring ports, the MRP_RingState shall be set to ring closed state.

Local variables of the MRM protocol machine are listed in Table 40.

Table 40 – MRP Local variables of MRM protocol machine

Name	Type	Meaning
SA_Port1	OctetString[6]	Ring port RPort_1 MAC source address
SA_Port2	OctetString[6]	Ring port RPort_2 MAC source address
SA_RPort	OctetString[6]	Ring port1 or Ring port2 MAC source address
PRIORITY	Unsigned8	Priority according to IEEE 802.1Q for MRP-PDU. Shall be set to ORG.
MRP_TS_Prio	Unsigned16	MRP_Prio of host
MRP_TS_SA	OctetString[6]	MAC source address of host
RPort_1	Unsigned16	Port identification of ring port 1
RPort_2	Unsigned16	Port identification of ring port 2
PRM_RPort	Unsigned16	Port identification of primary ring port
SEC_RPort	Unsigned16	Port identification of secondary ring port
MRP_MRM_NRmax	Unsigned16	Maximum retransmission count of MRP-PDU of Type MRP_Test
MRP_MRM_NReturn	Unsigned16	Counter, range MRP_MRM_NRmax to 0
TC_NReturn	Unsigned16	Counter, range MRP_TOPNRmax to 0
ADD_TEST	Boolean	Send additional MRP-PDU of type MRP_Test after MRP_TSTshortT intervall if TRUE
REACT_MODE	Boolean	MRM reacts on MRP_LinkDown frames from an MRC if TRUE
MRP_LNK_UP	Unsigned16	Constant value to indicate Link Up
MRP_LNK_DOWN	Unsigned16	Constant value to indicate Link Down
MRP_BLOCKED_SUPPORTED	Unsigned16	Constant value to indicate that – if TRUE – the MRM assumes all MRC in the ring support the BLOCKED port state If FALSE, the MRM requires additional support for MRC not supporting BLOCKED state FALSE: option for nodes not according to IEC61784-2, CP3/4, CP3/5, CP3/6.
MRP_REACT_ON_LINK_CHANGE	Unsigned16	Constant value to indicate that – if TRUE – the MRM reacts on MRP_LinkDown frames from an MRC with TopologyChange. If FALSE, the MRM does not react on MRP_LinkDown frames TRUE: option for nodes not according to IEC 61784-1, IEC 61784-2, CP3/4, CP3/5, CP3/6
NO_TC	Boolean	Suppress MRP_TopologyChange while in line topology (NO_TC = TRUE)

The MRM state machine shall be according to Table 41.

Table 41 – MRM State machine

#	Current state	Event /Condition =>Action	Next state
1	Power On	=> InitFDB() AddMACFDB({local},{MC_TEST, MC_CONTROL},ORG) AddMACFDB({local},{MC_INCONTROL},ORG) PRM_RPort:= RPort_1 SEC_RPort:= RPort_2 MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE REACT_MODE:= TRUE if MRM reacts on MRP_LinkDown frames from an MRC or FALSE if MRM does not react on MRP_LinkDown frames from an MRC Set_Port_StateReq (PRM_RPort, BLOCKED) Set_Port_StateReq (SEC_RPort, BLOCKED)	AC_STAT1
2	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT)	PRM_UP
3	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => SEC_RPort:= PRM_RPort PRM_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT)	PRM_UP
5	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
6	AC_STAT1	TestTimer expired => ignore	AC_STAT1

#	Current state	Event /Condition =>Action	Next state
7	AC_STAT1	LinkChangeInd(PortMode, Link_status) => ignore	AC_STAT1
8	PRM_UP	TestTimer expired => ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	PRM_UP
9	PRM_UP	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PRM_UP
10	PRM_UP	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => TestTimer.stop SetPortStateReq(PRM_RPort, BLOCKED)	AC_STAT1
11	PRM_UP	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	PRM_UP
12	PRM_UP	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= TRUE TestRingReq(MRP_TSTdefaultT)	CHK_RC
13	PRM_UP	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
14	PRM_UP	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA => ignore	PRM_UP

#	Current state	Event /Condition =>Action	Next state
15	PRM_UP	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && PortMode == MRP_BLOCKED_SUPPORTED => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	PRM_UP
16	PRM_UP	LinkChangeInd(PortMode, Link_status) /ADD_TEST && PortMode == MRP_BLOCKED_SUPPORTED => ignore	PRM_UP
17	PRM_UP	LinkChangeInd(PortMode, Link_status) /Link_status == MRP_LNK_DOWN && PortMode != MRP_BLOCKED_SUPPORTED => ignore	PRM_UP
18	PRM_UP	LinkChangeInd(PortMode, Link_status) /ADD_TEST && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => TopologyChangeReq(0)	PRM_UP
19	PRM_UP	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT) TopologyChangeReq(0)	PRM_UP
20	PRM_UP	TopologyChangeInd(MRP_SA, t) => ignore	PRM_UP
21	CHK_RO	TestTimer expired => ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RO
22	CHK_RO	MAUTypeChangeInd(RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RO

#	Current state	Event /Condition =>Action	Next state
23	CHK_RO	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq(SEC_RPort, BLOCKED) TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	PRM_UP
24	CHK_RO	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => Ignore	CHK_RO
25	CHK_RO	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq(SEC_RPort, BLOCKED)	PRM_UP
26	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA && REACT_MODE != MRP_REACT_ON_LINK_CHANGE => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	CHK_RC
27	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA && REACT_MODE == MRP_REACT_ON_LINK_CHANGE => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(0)	CHK_RC
28	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA => ignore	CHK_RO

#	Current state	Event /Condition =>Action	Next state
29	CHK_RO	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && Link_status == MRP_LNK_UP && PortMode == MRP_BLOCKED_SUPPORTED => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RO
30	CHK_RO	LinkChangeInd(PortMode, Link_status) /ADD_TEST && Link_status == MRP_LNK_UP && PortMode == MRP_BLOCKED_SUPPORTED => Ignore	CHK_RO
31	CHK_RO	LinkChangeInd(PortMode, Link_status) /ADD_TEST && Link_status == MRP_LNK_DOWN => Ignore	CHK_RO
32	CHK_RO	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && Link_status == MRP_LNK_DOWN => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RO
33	CHK_RO	LinkChangeInd(PortMode, Link_status) /ADD_TEST && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTExtNRmax – 1 MRP_MRM_NReturn:= 0 TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(0)	CHK_RC

#	Current state	Event /Condition =>Action	Next state
34	CHK_RO	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTExtNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT) TopologyChangeReq(0)	CHK_RC
35	CHK_RO	TopologyChangeInd(MRP_SA, t) => ignore	CHK_RO
36	CHK_RC	TestTimer expired /MRP_MRM_NReturn >= MRP_MRM_NRmax && !NO_TC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE TopologyChangeReq(MRP_TOPchgT) TestRingReq(MRP_TSTdefaultT)	CHK_RO
37	CHK_RC	TestTimer expired /MRP_MRM_NReturn >= MRP_MRM_Nrmax && NO_TC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RO
38	CHK_RC	TestTimer expired /MRP_MRM_NReturn < MRP_MRM_NRmax => MRP_MRM_NReturn:= MRP_MRM_NReturn + 1 ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
39	CHK_RC	MAUTypeChangeInd(RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Ignore	CHK_RC

#	Current state	Event /Condition =>Action	Next state
40	CHK_RC	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	PRM_UP
41	CHK_RC	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => Ignore	CHK_RC
42	CHK_RC	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => Ignore	PRM_UP
43	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE	CHK_RC
44	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA => ignore	CHK_RC
45	CHK_RC	LinkChangeInd(PortMode, Link_status) /ADD_TEST && REACT_MODE != MRP_REACT_ON_LINK_CHANGE && PortMode == MRP_BLOCKED_SUPPORTED => ignore	CHK_RC
46	CHK_RC	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && REACT_MODE != MRP_REACT_ON_LINK_CHANGE && PortMode == MRP_BLOCKED_SUPPORTED => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RC

#	Current state	Event /Condition =>Action	Next state
47	CHK_RC	LinkChangeInd(PortMode, Link_status) /Link_status == MRP_LNK_DOWN && REACT_MODE == MRP_REACT_ON_LINK_CHANGE => Set_Port_StateReq (SEC_RPort, FORWARDING) TopologyChangeReq(0)	CHK_RO
48	CHK_RC	LinkChangeInd(PortMode, Link_status) /Link_status == MRP_LNK_UP && REACT_MODE == MRP_REACT_ON_LINK_CHANGE && PortMode != MRP_BLOCKED_SUPPORTED => MRP_MRM_NRmax:= MRP_TSTExtNRmax – 1 TopologyChangeReq(0)	CHK_RC
49	CHK_RC	LinkChangeInd(PortMode, Link_status) /Link_status == MRP_LNK_UP && REACT_MODE == MRP_REACT_ON_LINK_CHANGE && PortMode == MRP_BLOCKED_SUPPORTED => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 TopologyChangeReq(0)	CHK_RC
50	CHK_RC	TopologyChangeInd(MRP_SA, t) => Ignore	CHK_RC
51	PRM_UP	InterconnTopologyChangeInd(MRP_SA, t, InID) /TopTimer.isRunning == FALSE => TopologyChangeReq(t)	PRM_UP
52	CHK_RO	InterconnTopologyChangeInd(MRP_SA, t, InID) /TopTimer.isRunning == FALSE => TopologyChangeReq(t)	CHK_RO
53	CHK_RO	InterconnLinkChangeInd(InID, LinkStatus, RPort) /RPort == PRM_RPort => InterconnForwardReq(SEC_RPort)	CHK_RO
54	CHK_RO	InterconnLinkChangeInd(InID, LinkStatus, RPort) /RPort == SEC_RPort => InterconnForwardReq(PRM_RPort)	CHK_RO
55	CHK_RO	InterconnLinkStatusInd(InID, RPort) /RPort == PRM_RPort => InterconnForwardReq(SEC_RPort)	CHK_RO

#	Current state	Event /Condition =>Action	Next state
56	CHK_RO	InterconnLinkStatusInd(InID, RPort) /RPort == SEC_RPort => InterconnForwardReq(PRM_RPort)	CHK_RO
57	CHK_RC	InterconnTopologyChangeInd(MRP_SA, t, InID) /TopTimer.isRunning == FALSE => TopologyChangeReq(t)	CHK_RC

8.2.2 MRC protocol machine

The MRC protocol machine is defined in Table 43. The principal behavior of the protocol machine is shown in Figure 12.

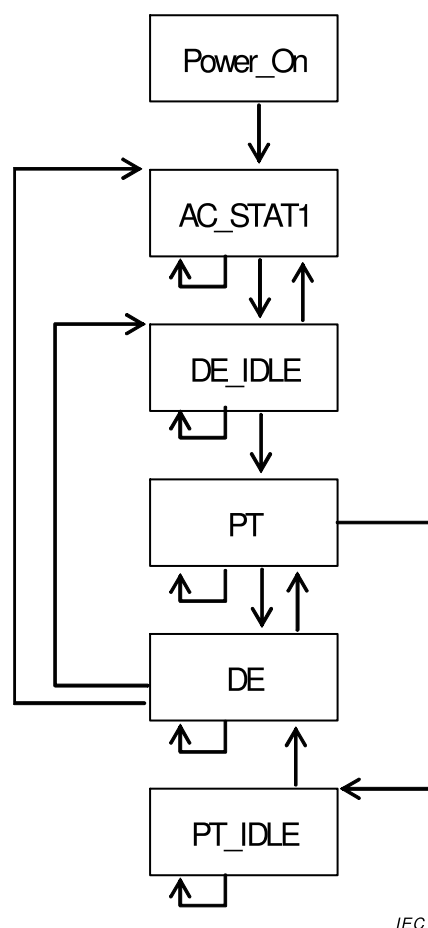


Figure 12 – MRP protocol machine for MRC

The text below is an explanation of the overall actions performed in the states. If a difference in the interpretation occurs between this text and the state machine, then the state machine supersedes.

Power_On

Initialization, the MRC shall start with both ring ports RPort_1 and RPort_2 in the port state BLOCKED. Static FDB entries for MRP multicast addresses MC_TEST and MC_CONTROL are generated: Forward MRP frames to MC_TEST and MC_CONTROL

between ring ports and frames to MC_CONTROL also to host. All MRP-PDU shall use the highest priority (ORG).

AC_STAT1 (Awaiting Connection State 1)

Startup, wait for Link Up on one of the ring ports.

DE_IDLE (Data Exchange Idle state)

This state shall be reached if only one ring port (primary) has a link and its port state is set to FORWARDING.

PT (Pass Through)

Temporary state while signaling link changes.

DE (Data Exchange)

Temporary state while signaling link changes.

PT_IDLE (Pass Through Idle state)

This state shall be reached if both ring ports have a link and their port states are set to FORWARDING.

Local variables of the MRC protocol machine are listed in Table 42.

Table 42 – MRP Local variables of MRC protocol machine

Name	Type	Meaning
SA_RPort	OctetString[6]	Ring port1 or ring port2 MAC source address
PRIORITY	Unsigned8	Priority according to IEEE 802.1Q for MRP-PDU. Shall be set to ORG.
RPort_1	Unsigned16	Port identification of ring port 1
RPort_2	Unsigned16	Port identification of ring port 2
PRM_RPort	Unsigned16	Port identification of primary ring port
SEC_RPort	Unsigned16	Port identification of secondary ring port
MRP_LNKNReturn	Unsigned16	Counter, Range MRP_LNKNRmax to 0
MRP_LNK_UP	Unsigned16	Constant value to indicate Link Up
MRP_LNK_DOWN	Unsigned16	Constant value to indicate Link Down

The MRC state machine shall be according to Table 43.

Table 43 – MRC state machine

#	Current state	Event /Condition =>action	Next state
1	Power On	=> InitFDB() AddMACFDB({RPort_1,RPort_2}, {MC_TEST, MC_CONTROL},ORG) AddMACFDB({local},{MC_CONTROL},ORG) PRM_RPort:= RPort_1 SEC_RPort:= RPort_2 Set_Port_StateReq (PRM_RPort, BLOCKED) Set_Port_StateReq (SEC_RPort, BLOCKED) UpTimer.ini DownTimer.ini	AC_STAT1
2	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Set_Port_StateReq (PRM_RPort, FORWARDING)	DE_IDLE
3	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => SEC_RPort:= PRM_RPort PRM_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING)	DE_IDLE
5	AC_STAT1	TopologyChangeInd (MRP_SA, t) => ignore	AC_STAT1
6	DE_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.start(MRP_LNKupT) LinkChangeReq(PRM_RPort, MRP_LNK_UP, MRP_LNKNReturn X MRP_LNKupT)	PT

#	Current state	Event /Condition =>action	Next state
7	DE_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE_IDLE
8	DE_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq (PRM_RPort, BLOCKED)	AC_STAT1
9	DE_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE_IDLE
10	DE_IDLE	TopologyChangeInd(MRP_SA, t) => ClearFDB(t)	DE_IDLE
11	PT	UpTimer expired /MRP_LNKReturn == 0 => MRP_LNKReturn:= MRP_LNKReturnmax Set_Port_StateReq (SEC_RPort, FORWARDING)	PT_IDLE
12	PT	UpTimer expired /MRP_LNKReturn > 0 => MRP_LNKReturn:= MRP_LNKReturn – 1 UpTimer.start(MRP_LNKupT) LinkChangeReq(PRM_RPort, MRP_LNK_UP, MRP_LNKReturn X MRP_LNKupT)	PT
13	PT	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT

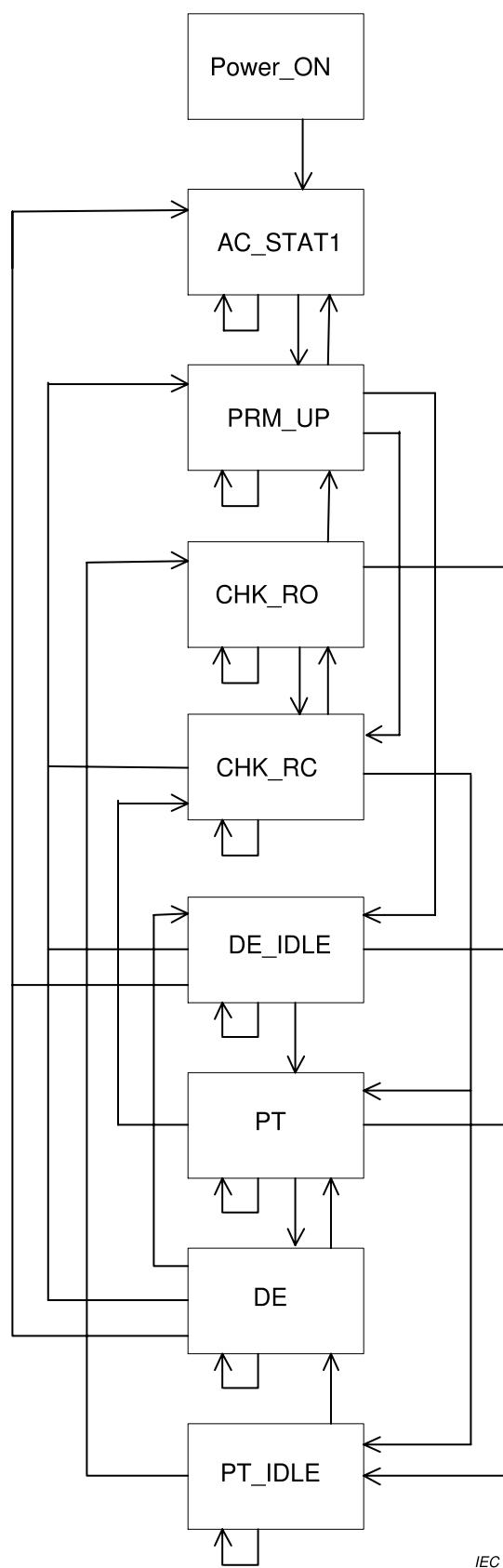
#	Current state	Event /Condition =>action	Next state
14	PT	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKReturn:= MRP_LNKmax UpTimer.stop Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort , MRP_LNK_DOWN, MRP_LNKReturn X MRP_LNKdownT)	DE
15	PT	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKReturn:= MRP_LNKmax UpTimer.stop PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING) Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort , MRP_LNK_DOWN, MRP_LNKReturn X MRP_LNKdownT)	DE
16	PT	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT
17	PT	TopologyChangeInd(MRP_SA, t) => MRP_LNKReturn:= MRP_LNKmax UpTimer.stop Set_Port_StateReq (SEC_RPort, FORWARDING) ClearFDB(t)	PT_IDLE
18	DE	DownTimer expired /MRP_LNKReturn == 0 => MRP_LNKReturn:= MRP_LNKmax	DE_IDLE

#	Current state	Event /Condition =>action	Next state
19	DE	DownTimer expired /MRP_LNKNReturn > 0 => MRP_LNKNReturn:= MRP_LNKNReturn – 1 DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn X MRP_LNKdownT)	DE
20	DE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_LNKNReturn:= MRP_LNKNRmax DownTimer.stop UpTimer.start(MRP_LNKupT) LinkChangeReq(PRM_RPort , MRP_LNK_UP, MRP_LNKNReturn X MRP_LNKupT)	PT
21	DE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE
22	DE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (PRM_RPort, BLOCKED) DownTimer.stop	AC_STAT1
23	DE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE
24	DE	TopologyChangeInd(MRP_SA, t) => MRP_LNKNReturn:= MRP_LNKNRmax DownTimer.stop ClearFDB(t)	DE_IDLE

#	Current state	Event /Condition =>action	Next state
25	PT_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
26	PT_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort , MRP_LNK_DOWN, MRP_LNKNReturn X MRP_LNKdownT)	DE
27	PT_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort , MRP_LNK_DOWN, MRP_LNKNReturn X MRP_LNKdownT)	DE
28	PT_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
29	PT_IDLE	TopologyChangeInd(MRP_SA, t) => ClearFDB(t)	PT_IDLE

8.2.3 MRA protocol machine

The MRA protocol machine is defined in Table 45. The principal behavior of the protocol machine is shown in Figure 13.



IEC

Figure 13 – MRP protocol machine for MRA

Basically the MRA protocol machine is a combination of the MRM protocol machine and the MRC protocol machine.

For an explanation of the states of the MRA protocol machine refer to the explanations of the MRM protocol machine in 8.2.1 and to the explanations of the MRC protocol state machine in 8.2.2. If a difference in the interpretation occurs between the text of the explanations and the state machine, then the state machine supersedes.

The MRA protocol machine uses the local variables of the MRM protocol machine and the local variables of the MRC protocol machine. In addition the MRA protocol machine uses further local variables as listed in Table 44.

Table 44 – MRP local variables of MRA protocol machine

Name	Type	Meaning
MRP_BestMRM_SA	OctetString[6]	MAC source address of the currently known recorded higher priority manager
MRP_BestMRM_Prio	Unsigned16	MRP_Prio of the currently known recorded higher priority manager
HO_BestMRM_SA	OctetString[6]	MAC source address of the recorded higher priority manager stored by host
HO_BestMRM_Prio	Unsigned16	MRP_Prio of the recorded higher prioritymanager stored by host
MRP_MonNRReturn	Unsigned16	Retry counter to monitor the recorded higher priority MRA
MRP_mySA	OctetString[6]	Host MAC source address
MRP_myPrio	Unsigned16	Host Priority
MRP_remotePrio	Unsigned16	Received Priority
LNKUP_HYST_TIMER_RUNNING	Unsigned16	Flag for linkup-hysterisis-timer: TRUE: Timer is running FALSE: Timer has elapsed or has not been started. The linkup-hysterisis-timer is serviced outside the state machine. It shall be started when the device recognizes a link-up and shall be stopped when the timer has elapsed or the device recognizes a link-down at the port. A MAUType_ChangeInd at link-up recognition shall first be launched, when the linkup-hysteresis-timer has elapsed.

The MRA state machine shall be according to Table 45.

Table 45 – MRA state machine

#	Current State	Event /Condition =>Action	Next State
1	Power On	=> InitFDB() AddMACFDB({local},{MC_TEST, MC_CONTROL},ORG) PRM_RPort:= RPort_1 SEC_RPort:= RPort_2 MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE REACT_MODE:= FALSE MRP_BestMRM_SA:= 0xFFFFFFFFFFFF MRP_BestMRM_Prio:= 0xFFFF Set_Port_StateReq (PRM_RPort, BLOCKED) Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MonNReturn:= 0	AC_STAT1
2	AC_STAT1	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT)	PRM_UP
3	AC_STAT1	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
4	AC_STAT1	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => SEC_RPort:= PRM_RPort PRM_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT)	PRM_UP
5	AC_STAT1	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
6	AC_STAT1	TestTimer expired => ignore	AC_STAT1

#	Current State	Event /Condition =>Action	Next State
7	AC_STAT1	LinkChangeInd(PortMode, Link_status) => ignore	AC_STAT1
8	PRM_UP	TestTimer expired => ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	PRM_UP
9	PRM_UP	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status ==MRP_LNK_UP => ignore	PRM_UP
10	PRM_UP	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => TestTimer.stop SetPortStateReq(PRM_RPort, BLOCKED)	AC_STAT1
11	PRM_UP	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort&& Link_status ==MRP_LNK_DOWN => ignore	PRM_UP
12	PRM_UP	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= TRUE TestRingReq(MRP_TSTdefaultT)	CHK_RC
13	PRM_UP	TestRingInd (MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
14	PRM_UP	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => ignore	PRM_UP

#	Current State	Event /Condition =>Action	Next State
15	PRM_UP	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => TestMgrNAckReq(MRP_Prio, MRP_SA)	PRM_UP
16	PRM_UP	LinkChangeInd(PortMode, Link_status) /!ADD_TEST => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	PRM_UP
17	PRM_UP	LinkChangeInd(PortMode, Link_status) /ADD_TEST => ignore	PRM_UP
18	PRM_UP	TopologyChangeInd(MRP_SA, t) /MRP_SA == MRP_TS_SA => ignore	PRM_UP
19	PRM_UP	TopologyChangeInd(MRP_SA, t) /MRP_SA != MRP_TS_SA => CLEAR_FDB(t)	PRM_UP
20	PRM_UP	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA == MRP_TS_SA => ignore	PRM_UP
21	PRM_UP	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA != MRP_TS_SA => ignore	PRM_UP
22	PRM_UP	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA ==MRP_TS_SA && !REM_PRIO_BETTER_THAN_HO_PRIO => TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	DE_IDLE

#	Current State	Event /Condition =>Action	Next State
23	PRM_UP	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && REM_PRIO_BETTER_THAN_HO_PRIO => HO_BestMRM_SA:= MRP_SA HO_BestMRM_Prio:= MRP_Prio TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	DE_IDLE
24	PRM_UP	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) => ignore	PRM_UP
25	CHK_RO	TestTimer expired => ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RO
26	CHK_RO	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_Rport && Link_status == MRP_LNK_UP => ignore	CHK_RO
27	CHK_RO	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => PRM_RPort:= SEC_RPort SEC_RPort:= Rport Set_Port_StateReq(SEC_RPort, BLOCKED) TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	PRM_UP
28	CHK_RO	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_Rport && Link_status == MRP_LNK_UP => ignore	CHK_RO
29	CHK_RO	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq(SEC_RPort, BLOCKED)	PRM_UP

#	Current State	Event /Condition =>Action	Next State
30	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	CHK_RC
31	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => ignore	CHK_RO
32	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => TestMgrNAckReq(MRP_Prio, MRP_SA)	CHK_RO
33	CHK_RO	LinkChangeInd(PortMode, LinkStatus) /!ADD_TEST => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RO
34	CHK_RO	LinkChangeInd(PortMode, LinkStatus) /ADD_TEST => ignore	CHK_RO
35	CHK_RO	TopologyChangeInd(MRP_SA, t) /MRP_SA == MRP_TS_SA => ignore	CHK_RO
36	CHK_RO	TopologyChangeInd(MRP_SA, t) /MRP_SA != MRP_TS_SA => CLEAR_FDB(t)	CHK_RO
37	CHK_RO	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA == MRP_TS_SA => ignore	CHK_RO
38	CHK_RO	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA != MRP_TS_SA => ignore	CHK_RO

#	Current State	Event /Condition =>Action	Next State
39	CHK_RO	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && !REM_PRIO_BETTER_THAN_HO_PRIO => TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	PT_IDLE
40	CHK_RO	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && REM_PRIO_BETTER_THAN_HO_PRIO => O_BestMRM_SA:= MRP_SA HO_BestMRM_Prio:= MRP_Prio TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	PT_IDLE
41	CHK_RO	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) => ignore	CHK_RO
42	CHK_RC	TestTimer expired /MRP_MRM_NReturn >= MRP_MRM_Nrmax && !NO_TC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE TopologyChangeReq(MRP_TOPchgT) TestRingReq(MRP_TSTdefaultT)	CHK_RO
43	CHK_RC	TestTimer expired /MRP_MRM_NReturn >= MRP_MRM_Nrmax&& NO_TC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE TopologyChangeReq(MRP_TOPchgT) TestRingReq(MRP_TSTdefaultT)	CHK_RO

#	Current State	Event /Condition =>Action	Next State
44	CHK_RC	TestTimer expired /MRP_MRM_NReturn < MRP_MRM_Nrmax => MRP_MRM_NReturn:= MRP_MRM_NReturn + 1 ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
45	CHK_RC	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RC
46	CHK_RC	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	PRM_UP
47	CHK_RC	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RC
48	CHK_RC	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	PRM_UP
49	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE	CHK_RC
50	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => TestMgrNAckReq(MRP_Prio, MRP_SA)	CHK_RC
51	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => ignore	CHK_RC

#	Current State	Event /Condition =>Action	Next State
52	CHK_RC	LinkChangeInd(PortMode, Link_status) /ADD_TEST => ignore	CHK_RC
53	CHK_RC	LinkChangeInd(PortMode, Link_status) /!ADD_TEST =>ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RC
54	CHK_RC	TopologyChangeInd(MRP_SA, t) /MRP_SA == MRP_TS_SA => ignore	CHK_RC
55	CHK_RC	TopologyChangeInd(MRP_SA, t) /MRP_SA != MRP_TS_SA => CLEAR_FDB(t)	CHK_RC
56	CHK_RC	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA == MRP_TS_SA => ignore	CHK_RC
57	CHK_RC	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA != MRP_TS_SA => ignore	CHK_RC
58	CHK_RC	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && !REM_PRIO_BETTER_THAN_HO_PRIO => Set_Port_StateReq (SEC_RPort, FORWARDING) TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	PT_IDLE
59	CHK_RC	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && REM_PRIO_BETTER_THAN_HO_PRIO => HO_BestMRM_SA:= MRP_SA HO_BestMRM_Prio:= MRP_Prio TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	PT_IDLE

#	Current State	Event /Condition =>Action	Next State
60	CHK_RC	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) => ignore	CHK_RC
61	DE_IDLE	TestTimer expired /MRP_MonNReturn <= MRP_MON_NR_MAX => MRP_MonNReturn:= MRP_MonNReturn + 1 TestTimer.start(MRP_TSTshortT)	DE_IDLE
62	DE_IDLE	TestTimer expired /MRP_MonNReturn > MRP_MON_NR_MAX => TestTimer.start(MRP_TSTshortT) MRM_Init()	PRM_UP
63	DE_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.start(MRP_LNKupT) AddMACFDB({PRM_RPort, SEC_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_Rport, MRP_LNK_UP, MRP_LNKNReturn * MRP_LNKupT)	PT
64	DE_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE_IDLE
65	DE_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq (PRM_RPort, BLOCKED) AddMACFDB({}, {MC_TEST, MC_CONTROL},ORG)	AC_STAT1
66	DE_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE_IDLE
67	DE_IDLE	TopologyChangeInd(MRP_SA, t) => CLEAR_FDB(t)	DE_IDLE

#	Current State	Event /Condition =>Action	Next State
68	DE_IDLE	TopologyChangeInd(MRP_SA, t) /LNKUP_HYST_TIMER_RUNNING == TRUE => CLEAR_FDB(t) AddMACFDB({PRM_RPort,SEC_RPort}, {MC_TEST, MC_CONTROL},ORG) Set_Port_StateReq (SEC_RPort,FORWARDING)	PT_IDLE
69	DE_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => ignore	DE_IDLE
70	DE_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	DE_IDLE
71	DE_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA &&MRP_SA == HO_BestMRM_SA &&!REM_PRIO_BETTER_THAN_OWN_PRIO => HO_BestMRM_Prio:= MRP_Prio	DE_IDLE
72	DE_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA &&MRP_SA == HO_BestMRM_SA &&REM_PRIO_BETTER_THAN_OWN_PRIO => MRP_MonNRReturn:= 0 HO_BestMRM_Prio:= MRP_Prio	DE_IDLE
73	DE_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA == MRP_TS_SA => ignore	DE_IDLE
74	DE_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	DE_IDLE
75	DE_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == MRP_BestMRM_SA => HO_BestMRM_SA:= MRP_BestMRM_SA HO_BestMRM_Prio:= MRP_BestMRM_Prio MRP_MonNRReturn:= 0	DE_IDLE

#	Current State	Event /Condition =>Action	Next State
76	DE_IDLE	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) => ignore	DE_IDLE
77	DE_IDLE	LinkChangeInd(PortMode, Link_status) => ignore	DE_IDLE
78	PT	TestTimer expired /MRP_MonNReturn <= MRP_MON_NR_MAX => MRP_MonNReturn:= MRP_MonNReturn + 1 TestTimer.start(MRP_TSTshortT)	PT
79	PT	TestTimer expired /MRP_MonNReturn > MRP_MON_NR_MAX => TestTimer.start(MRP_TSTshortT) MRM_Init()	CHK_RC
80	PT	UpTimer expired /MRP_LNKNReturn == 0 => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (SEC_RPort, FORWARDING)	PT_IDLE
81	PT	UpTimer expired /MRP_LNKNReturn > 0 => MRP_LNKNReturn:= MRP_LNKNReturn -1 UpTimer.start(MRP_LNKupT) LinkChangeReq(PRM_RPort, MRP_LNK_UP, MRP_LNKNReturn * MRP_LNKupT)	PT
82	PT	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT
83	PT	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.stop Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) AddMACFDB({PRM_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE

#	Current State	Event /Condition =>Action	Next State
84	PT	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.stop PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING) Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) AddMACFDB({PRM_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE
85	PT	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT
86	PT	TopologyChangeInd(MRP_SA, t) => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.stop Set_Port_StateReq (SEC_RPort, FORWARDING) CLEAR_FDB(t)	PT_IDLE
87	PT	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => ignore	PT
88	PT	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	PT
89	PT	TestRingInd(MRP_SA, MRP_Prio /MRP_SA != MRP_TS_SA && MRP_SA == HO_BestMRM_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => HO_BestMRM_Prio:= MRP_Prio	PT

#	Current State	Event /Condition =>Action	Next State
90	PT	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == HO_BestMRM_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => MRP_MonNReturn:= 0 HO_BestMRM_Prio:= MRP_Prio	PT
91	PT	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA == MRP_TS_SA => ignore	PT
92	PT	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	PT
93	PT	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == MRP_BestMRM_SA => HO_BestMRM_SA:= MRP_BestMRM_SA HO_BestMRM_Prio:= MRP_BestMRM_Prio MRP_MonNReturn:= 0	PT
94	PT	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) => ignore	PT
95	PT	LinkChangeInd(PortMode, Link_status) => ignore	PT
96	DE	TestTimer expired /MRP_MonNReturn <= MRP_MON_NR_MAX => MRP_MonNReturn:= MRP_MonNReturn + 1 TestTimer.start(MRP_TSTdefaultT)	DE
97	DE	TestTimer expired /MRP_MonNReturn > MRP_MON_NR_MAX => TestTimer.start(MRP_TSTshortT) MRM_Init()	PRM_UP
98	DE	DownTimer expired /MRP_LNKNReturn == 0 => MRP_LNKNReturn:= MRP_LNKNRmax	DE_IDLE

#	Current State	Event /Condition =>Action	Next State
99	DE	DownTimer expired /MRP_LNKNReturn > 0 => MRP_LNKNReturn:= MRP_LNKNReturn – 1 DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE
100	DE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_LNKNReturn:= MRP_LNKNRmax DownTimer.stop UpTimer.start(MRP_LNKupT) AddMACFDB({PRM_RPort, SEC_RPort}, {MC_TEST, MC_CONTROL},ORG LinkChangeReq(PRM_RPort, MRP_LNK_UP, MRP_LNKNReturn * MRP_LNKupT)	PT
101	DE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE
102	DE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (PRM_RPort, BLOCKED) DownTimer.stop AddMACFDB({}, {MC_TEST, MC_CONTROL},ORG)	AC_STAT1
103	DE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE
104	DE	TopologyChangeInd(MRP_SA, t) => MRP_LNKNReturn:= MRP_LNKNRmax DownTimer.stop CLEAR_FDB(t)	DE_IDLE
105	DE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => ignore	DE

#	Current State	Event /Condition =>Action	Next State
106	DE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	DE
107	DE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == HO_BestMRM_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => HO_BestMRM_Prio:= MRP_Prio	DE
108	DE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == HO_BestMRM_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => MRP_MonNReturn:= 0 HO_BestMRM_Prio:= MRP_Prio	DE
109	DE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA == MRP_TS_SA => ignore	DE
110	DE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	DE
111	DE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == MRP_BestMRM_SA => HO_BestMRM_SA:= MRP_BestMRM_SA HO_BestMRM_Prio:= MRP_BestMRM_Prio MRP_MonNReturn:= 0	DE
112	DE	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) => ignore	DE
113	DE	LinkChangeInd(PortMode, Link_status) => ignore	DE

#	Current State	Event /Condition =>Action	Next State
114	PT_IDLE	TestTimer expired /MRP_MonNReturn <= MRP_MON_NR_MAX => MRP_MonNReturn:= MRP_MonNReturn + 1 TestTimer.start(MRP_TSTdefaultT)	PT_IDLE
115	PT_IDLE	TestTimer expired /MRP_MonNReturn > MRP_MON_NR_MAX => TestTimer.start(MRP_TSTshortT) MRM_Init()	CHK_RO
116	PT_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
117	PT_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) AddMACFDB({PRM_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE
118	PT_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) AddMACFDB({PRM_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE
119	PT_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
120	PT_IDLE	TopologyChangeInd(MRP_SA, t) => CLEAR_FDB(t)	PT_IDLE

#	Current State	Event /Condition =>Action	Next State
121	PT_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => ignore	PT_IDLE
122	PT_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	PT_IDLE
123	PT_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA &&MRP_SA == HO_BestMRM_SA &&!REM_PRIO_BETTER_THAN_OWN_PRIO => HO_BestMRM_Prio:= MRP_Prio	PT_IDLE
124	PT_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA MRP_SA == HO_BestMRM_SA REM_PRIO_BETTER_THAN_OWN_PRIO => MRP_MonNReturn:= 0 HO_BestMRM_Prio:= MRP_Prio	PT_IDLE
125	PT_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA == MRP_TS_SA => ignore	PT_IDLE
126	PT_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	PT_IDLE
127	PT_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == MRP_BestMRM_SA => HO_BestMRM_SA:= MRP_BestMRM_SA HO_BestMRM_Prio:= MRP_BestMRM_Prio MRP_MonNReturn:= 0	PT_IDLE
128	PT_IDLE	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) => ignore	PT_IDLE

#	Current State	Event /Condition =>Action	Next State
129	PT_IDLE	LinkChangeInd(PortMode, Link_status) => ignore	PT_IDLE

8.2.4 MRA, MRM and MRC functions

The MRA, MRM and MRC functions shall be according to Table 46.

Table 46 – MRP functions and macros

Function name	Operations
TestRingReq(t)	SetupTestRingReq() TestTimer.start(t)
SetupTestRingReq()	Create MRP-PDU according MRP_Test Assignments: MRP_Type:= MRP_Test MRP_Prio:= MRP_TS_Prio MRP_SA:= MRP_TS_SA MRP_PortRole:= frame sent on primary ring port or secondary ring port MRP_RingState:= actual ring state MRP_Transition:= actual number of transitions between ring open state and ring closed state MRP_TimeStamp:= actual local counter value MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID If MRA MRP_Suboption1 = vendorspecific MRP_Suboption2 = MRP_AutoMgr MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_TEST, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_TEST, SA_Port2, PRIORITY, LT, MRP-PDU)
TestRingInd(MRP_SA, MRP_Prio)	Receive MRP-PDU according MRP_Test MRP_SA:= MRP_SA from MRP_PDU MRP_Prio:= MRP_Prio from MRP-PDU
TopologyChangeReq(time)	SetupTopologyChangeReq(MRP_TOPNRmax X time) if time == 0 ClearLocalFDB() else TopTimer.start(MRP_TOPchgT) TopTimer.isRunning = TRUE

Function name	Operations
SetupTopologyChangeReq(t)	Create MRP-PDU according MRP_TopologyChange Assignments: MRP_Type:= MRP_TopologyChange MRP_Prio:= MRP_TS_Prio MRP_SA:= MRP_TS_SA MRP_Interval:= t MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_CONTROL, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_CONTROL, SA_Port2, PRIORITY, LT, MRP-PDU)
TopologyChangeInd(MRP_SA, t)	Receive MRP-PDU according MRP_TopologyChange MRP_SA:= MRP_SA from MRP_PDU t:= MRP_Interval from MRP-PDU
LinkChangeReq(RPort, LinkStatus, time)	Create MRP-PDU according MRP_LinkUp or MRP_LinkDown Assignments: if LinkStatus == MRP_LNK_UP MRP_Type:= MRP_LinkUp else MRP_Type:= MRP_LinkDown MRP_SA:= MRP_TS_SA MRP_PortRole:= actual port role of the port which indicated the link change MRP_Interval:= time MRP_Blocked:= indicates if the MRC is able or is not able to receive and forward MRP_Test frames, MRP_LinkChange frames and MRP_TopologyChange frames on a ring port whose port state is BLOCKED. MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort, MC_CONTROL, SA_RPort, PRIORITY, LT, MRP-PDU)
LinkChangeInd(PortMode, LinkStatus)	Receive MRP-PDU according MRP_LinkDown or MRP_LinkUp PortMode:= MRP_Blocked from MRP-PDU if MRP_Type == MRP_LinkUp LinkStatus:= MRP_LNK_UP else LinkStatus:= MRP_LNK_DOWN

Function name	Operations
MAUTypeChangeInd(RPort, Link_status)	Receive a local link change indication. RPort:= port which caused the local link change indication. Link_status:= MRP_LNK_UP or MRP_LNK_DOWN (depends on the local link change indication)
SetPortStateReq(RPort, Status)	Function to set the port status of RPort to Status
ClearFDB(time)	FDBClearTimer.start(time)
ClearLocalFDB()	Function to clear the FDB within the MRP node. The learning of source addresses from ingress frames, which were sent out before the topology change was indicated, shall be prevented
InitFDB()	Function to initialize Filtering Database
AddMACFDB(Destination, MAC address, Priority)	Function to add Static Filtering Entries (MAC address) in the FDB with Priority and Destination. The term local in the state diagram means a connection to the Higher-Layer Entity (see IEEE 802.1D).
SendFrameReq(RPort, DestinationAddress, SourceAddress, Priority, LT, MRP-PDU)	Function to send an MRP-PDU at port RPort with the SourceAddress and LT to the DestinationAddress. Priority used in the TagControlInformation is coded in the frame if VLAN is used
TestMgrNAckReq(MRP_Prio, MRP_SA)	Create MRP-PDU according to MRP_Option , subtype MRP_TestMgrNAck Assignments: MRP_Type:= MRP_Option MRP_OUI:= ManufacturerOUI or IECOU MRP_SubOption1: manufacturer specific MRP_SubOption2: MRP_SubType:= SubTLVHeader.Type (= 0x01) MRP_Prio:= MRP_TS_Prio MRP_SA:= MRP_TS_SA MRP_OtherMRMPrio:= empty (0x00) MRP_OtherMRMSA:= MRP_SA of the remote MRA MRP_Type:= MRP_Common MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID SendFrameReq (RPort_1, MC_TEST, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_TEST, SA_Port2, PRIORITY, LT, MRP-PDU)

Function name	Operations
TestPropagateReq(MRP_Prio, MRP_SA)	Create MRP-PDU according to MRP_Option, subtype MRP_TestPropagate Assignments: MRP_Type:= MRP_Option MRP_OUI:= ManufacturerOUI or IECOU MRP_SubOption1: manufacturer specific MRP_SubOption2: MRP_SubType:= SubTLVHeader.Type (= 0x02) MRP_Prio:= MRP_TS_Prio MRP_SA:= MRP_TS_SA MRP_OtherMRMPrio:= MRP_Prio of the remote MRA who has higher priority MRP_OtherMRMSA:= MRP_SA of the remote MRA who has higher priority MRP_Type:= MRP_Common MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID SendFrameReq (RPort_1, MC_TEST, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_TEST, SA_Port2, PRIORITY, LT, MRP-PDU)
TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA)	Receive MRP-PDU according to MRP_Option, Subtype MRP_TestMgrNAck RPort:= R_Port MRP_SA:= MRP_SA from MRP-PDU MRP_Prio:= MRP_Prio from MRP-PDU MRP_BestMRM_SA:= MRP_SA from MRP-PDU
TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio)	Receive MRP-PDU according to MRP_Option, Subtype MRP_TestPropagateInd RPort:= R_Port MRP_SA:= MRP_SA from MRP-PDU MRP_Prio:= MRP_Prio from MRP-PDU MRP_BestMRM_SA:= MRP_OtherMRMSA from MRP-PDU MRP_BestMRM_Prio:= MRP_OtherMRMPrio from MRP-PDU
MRC_Init()	MRP_MonNReturn:= 0 MRP_LNKNReturn:= MRP_LNKNRmax AddMACFDB({RPort_1,RPort_2},{MC_TEST, MC_CONTROL},ORG) MRP_BestMRM_SA:= HO_BestMRM_SA MRP_BestMRM_Prio:= HO_BestMRM_Prio
MRM_Init()	AddTest:= FALSE NO_TC:= FALSE MRP_MRM_NReturn:= 0 AddMACFDB({}, {MC_TEST, MC_CONTROL},ORG) MRP_BestMRM_SA:= MRP_TS_SA MRP_BestMRM_Prio:= MRP_TS_Prio

Function name	Operations
REM_Prio_BETTER_THAN_OWN_Prio	Macro to check whether the priority of the remote device is higher than own priority? $(MRP_remotePrio < MRP_myPrio) \vee ((MRP_remotePrio == MRP_myPrio) \wedge (MRP_remoteSA < MRP_mySA))$ TRUE:= remote device has higher priority FALSE:= local host has higher priority
REM_Prio_BETTER_THAN_HO_Prio	Macro to check whether the priority of the remote device is higher than the higher priority managers priority, recorded by own host. $(MRP_remotePrio < HO_BestMRM_Prio) \vee ((MRP_remotePrio == HO_BestMRM_Prio) \wedge (MRP_remoteSA < HO_BestMRM_SA))$ TRUE:= remote device has higher priority FALSE:= recorded higher priority managers priority is higher

8.2.5 FDB clear timer

FDB clear timer is an auxiliary state machine. FDB clear timer shall be according to the state machine in Table 47.

Table 47 – MRP FDB clear timer

#	Current state	Event /Condition =>Action	Next state
1	Power On	=> FDBCclearTimer.ini	IDLE
2	IDLE	FDBCclearTimer .expired => ClearLocalFDB()	IDLE

8.2.6 Topology change timer

Topology change timer is an auxiliary state machine. Topology change timer shall be according to the state machine in Table 48.

Table 48 – MRP topology change timer

#	Current state	Event /Condition =>Action	Next state
1	Power On	=> TopTimer.ini TC_NReturn:= MRP_TOPNRmax – 1 TopTimer.isRunning = FALSE	IDLE
2	IDLE	TopTimer expired /TC_NReturn > 0 => SetupTopologyChangeReq(TC_NReturn X MRP_TOPchgT) TC_NReturn:= TC_NReturn – 1 TopTimer.start(MRP_TOPchgT)	IDLE
3	IDLE	TopTimer expired /TC_NReturn == 0 => TC_NReturn:= MRP_TOPNRmax – 1 CLEAR_ FDB(0) SetupTopologyChangeReq(0) TopTimer.isRunning = FALSE	IDLE

8.2.7 MIM protocol machine

The MIM protocol machine for LC-mode is defined in Table 50 and the MIM protocol machine for RC-mode is defined in Table 51. The principal behavior of both protocol machines is shown in Figure 14.

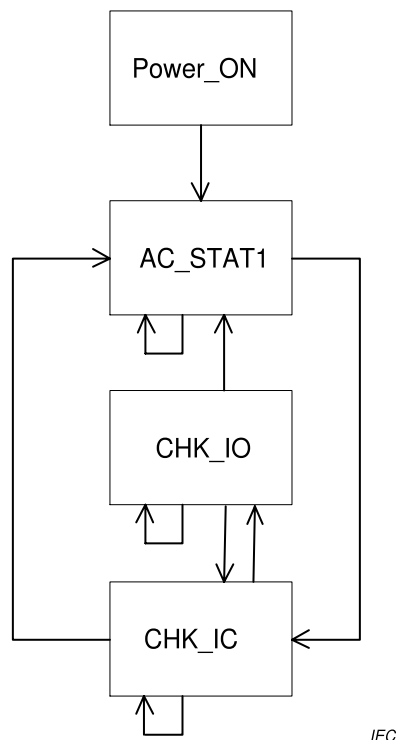


Figure 14 – MRP protocol machine for MIM in RC-mode and LC-mode

The text below is an explanation of the overall actions performed in the states. If a difference in the interpretation occurs between this text and the state machine, then the state machine supersedes.

PowerOn

Initialization, the MIM shall start with the interconnection port IPort in the port state BLOCKED. Static FDB entries for MRP multicast addresses are generated:

- In RC-mode: MC_INTEST and MC_INCONTROL. Forward MRP frames to MC_INTEST to host.
- In LC-mode: MC_INCONTROL.

Forward MRP frames to MC_INCONTROL between ring ports and to host. All MRP-PDU shall use the highest priority (ORG).

AC_STAT1 (Awaiting Connection State 1)

Startup, waiting for the Link Up at the interconnection port, and starting test monitoring of the interconnection if in RC_MODE.

CHK_IO (Check Interconnection, Interconnection Open State)

The MIM has a link up at its interconnection port and shall set its MRP_InState to interconnection open state when:

- In RC-mode: The MIM did not receive its MRP_InTest frames for a determined time or it received an MRP_InLinkDown.
- In LC-mode: The MIM received an MRP_InLinkDown frame

CHK_IC (Check Interconnection, Interconnection Closed State)

The MIM has a link up at its interconnection port and shall set its MRP_InState to interconnection closed state when:

- In RC-mode: The MIM receives its MRP_InTest frames
- In LC-mode: The MIM received an MRP_InLinkUp frame from an MIC which Interconnection ID is equal to the MIMs Interconnection ID.

Local variables of the MIM protocol machine are listed in Table 49.

Table 49 – MRP Local variables of MIM protocol machine

Name	Type	Meaning
RPort_1	Unsigned16	Port identification of ring port 1
RPort_2	Unsigned16	Port identification of ring port 2
IPort	Unsigned16	Port identification of interconnection port
SA_Port1	OctetString[6]	Ring port RPort_1 MAC source address
SA_Port2	OctetString[6]	Ring port RPort_2 MAC source address
SA_IPort	OctetString[6]	Interconnection port IPort MAC source address
PRIORITY	Unsigned8	Priority according to IEEE 802.1Q for MRP-PDU. Shall be set to ORG.
MRP_TS_SA	OctetString[6]	MAC source address of host
MRP_MIM_NRmax	Unsigned16	Maximum retransmission count of MRP-PDU of Type MRP_InTest
MRP_MIM_NReturn	Unsigned16	Counter, range MRP_MIM_NRmax to 0
IN_TC_NReturn	Unsigned16	Counter, range MRP_IN_TOPNRmax to 0
MRP_IN_LNKSTAT_NReturn	Unsigned16	Counter, range MRP_IN_LNKSTATNRmax to 0
IID	Unsigned16	Identification of interconnection domain
MRP_LNK_UP	Unsigned16	Constant value to indicate Link Up
MRP_LNK_DOWN	Unsigned16	Constant value to indicate Link Down

The MIM state machine for LC-mode shall be according to Table 50.

Table 50 – MIM State machine for LC-mode

#	Current state	Event /Condition =>Action	Next state
1	PowerOn	=> InitFDB() AddMACFDB({RPort_1,RPort_2}, {MC_INCONTROL},ORG) AddMACFDB({local},{MC_INCONTROL}, ORG) SetPortStateReq (IPort, BLOCKED) IID:= interconnection identifier	AC_STAT1
2	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => SetPortStateReq(IPort, BLOCKED) InterconnLinkStatusPollReq (MRP_IN_LNKSTATchgT)	CHK_IC
3	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => Ignore	AC_STAT1
4	AC_STAT1	InterconnLinkChangeInd(InID, Link_status, RPort) => Ignore	AC_STAT1
5	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	AC_STAT1
6	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	AC_STAT1
7	CHK_IO	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => Ignore	CHK_IO
8	CHK_IO	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, BLOCKED) InterconnLinkStatusTimer.stop	AC_STAT1

#	Current state	Event /Condition =>Action	Next state
9	CHK_IO	InterconnLinkChangeInd (InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_UP => SetPortStateReq (IPort, BLOCKED) InterconnLinkStatusTimer.stop InterconnTopologyChangeReq (MRP_IN_TOPchgT)	CHK_IC
10	CHK_IO	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_DOWN => SetPortStateReq (IPort, FORWARDING)	CHK_IO
11	CHK_IO	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	CHK_IO
12	CHK_IO	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	CHK_IO
13	CHK_IC	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => Ignore	CHK_IC
14	CHK_IC	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, BLOCKED)	AC_STAT1
15	CHK_IC	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, FORWARDING) InterconnLinkStatusTimer.stop InterconnTopologyChangeReq(MRP_IN_TOPchgT)	CHK_IO
16	CHK_IC	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_UP => Ignore	CHK_IC

#	Current state	Event /Condition =>Action	Next state
17	CHK_IC	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	CHK_IC
18	CHK_IC	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	CHK_IC

The MIM state machine for RC-mode shall be according to Table 51.

Table 51 – MIM State machine for RC-mode

#	Current state	Event /Condition =>Action	Next state
1	PowerOn	=> InitFDB() AddMACFDB({RPort_1,RPort_2}, {MC_INCONTROL},ORG) AddMACFDB({local},{MC_INCONTROL, MC_INTEST}, ORG) SetPortStateReq (IPort, BLOCKED)InterconnTestTimer.ini IID:= interconnection identifier	AC_STAT1
2	AC_STAT1	InterconnTestTimer expired => Ignore	AC_STAT1
3	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => SetPortStateReq(IPort, BLOCKED) MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0 InterconnTestReq(MRP_IN_TSTdefaultT)	CHK_IC
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => Ignore	AC_STAT1
5	AC_STAT1	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA == MRP_TS_SA => SetPortStateReq(IPort, BLOCKED) MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0 InterconnTestReq(MRP_IN_TSTdefaultT)	CHK_IC
6	AC_STAT1	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_1 => InterconnForwardReq(RPort_2)	AC_STAT1
7	AC_STAT1	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_2 => InterconnForwardReq(RPort_1)	AC_STAT1

#	Current state	Event /Condition =>Action	Next state
8	AC_STAT1	InterconnLinkChangeInd(InID, Link_status, RPort) => Ignore	AC_STAT1
9	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	AC_STAT1
10	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	AC_STAT1
11	CHK_IO	InterconnTestTimer expired => InterconnTestReq(MRP_IN_TSTdefaultT)	CHK_IO
12	CHK_IO	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => Ignore	CHK_IO
13	CHK_IO	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, BLOCKED) InterconnTopologyChangeReq(MRP_IN_TOPchgT) InterconnTestReq(MRP_IN_TSTdefaultT)	AC_STAT1
14	CHK_IO	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA == MRP_TS_SA => SetPortStateReq (IPort, BLOCKED) MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0 InterconnTopologyChangeReq (MRP_IN_TOPchgT) InterconnTestReq (MRP_IN_TSTdefaultT)	CHK_IC
15	CHK_IO	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_1 => InterconnForwardReq(RPort_2)	CHK_IO
16	CHK_IO	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_2 => InterconnForwardReq(RPort_1)	CHK_IO

#	Current state	Event /Condition =>Action	Next state
17	CHK_IO	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_UP => InterconnTestReq(MRP_IN_TSTdefaultT)	CHK_IO
18	CHK_IO	InterconnLinkChangeInd(InID, Link_status, RPort) /Link_status == MRP_LNK_DOWN => Ignore	CHK_IO
19	CHK_IO	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	CHK_IO
20	CHK_IO	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	CHK_IO
21	CHK_IC	InterconnTestTimer expired /MRP_MIM_NReturn >= MRP_MIM_NRmax => SetPortStateReq (IPort, FORWARDING) MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0 InterconnTopologyChangeReq (MRP_IN_TOPchgT) InterconnTestReq (MRP_IN_TSTdefaultT)	CHK_IO
22	CHK_IC	InterconnTestTimer expired /MRP_MIM_NReturn < MRP_MIM_NRmax => MRP_MIM_NReturn:= MRP_MIM_NReturn + 1 InterconnTestReq (MRP_IN_TSTdefaultT)	CHK_IC
23	CHK_IC	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => Ignore	CHK_IC
24	CHK_IC	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, BLOCKED) InterconnTopologyChangeReq(MRP_IN_TOPchgT) InterconnTestReq(MRP_IN_TSTdefaultT)	AC_STAT1

#	Current state	Event /Condition =>Action	Next state
25	CHK_IC	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA == MRP_TS_SA => MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0	CHK_IC
26	CHK_IC	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_1 => InterconnForwardReq(RPort_2)	CHK_IC
27	CHK_IC	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_2 => InterconnForwardReq(RPort_1)	CHK_IC
28	CHK_IC	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, FORWARDING) InterconnTopologyChangeReq(MRP_IN_TOPchgT)	CHK_IO
29	CHK_IC	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_UP => MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 InterconnTopologyChangeReq (MRP_IN_TOPchgT)	CHK_IC
30	CHK_IC	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	CHK_IC
31	CHK_IC	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	CHK_IC

8.2.8 MIC protocol machine

The MIC protocol machine for LC-mode is defined in Table 53 and the MIC protocol machine for RC-mode is defined in Table 54. The principal behavior of the protocol machine is shown in Figure 15.

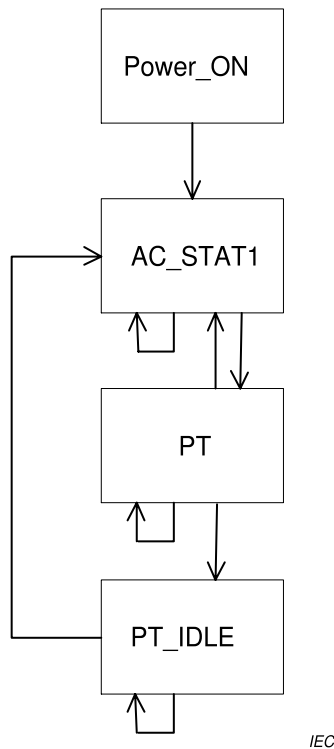


Figure 15 – MRP protocol machine for MIC in RC-mode and LC-mode

The text below is an explanation of the overall actions performed in the states. If a difference in the interpretation occurs between this text and the state machine, then the state machine supersedes.

PowerOn

Initialization, the MIC shall start with the interconnection port IPort in the port state BLOCKED. Static FDB entries for MRP multicast addresses are generated:

- In RC-mode: MC_INTEST and MC_INCONTROL. Forward MRP frames to MC_INTEST between ring ports and to the interconnection port.
- In LC-mode: MC_INCONTROL.

Forward MRP frames to MC_INCONTROL in both modes between ring ports and to host. All MRP-PDU shall use the highest priority (ORG).

AC_STAT1 (Awaiting Connection State 1)

Startup, waiting for the Link Up at the interconnection port.

PT (Pass Through)

Temporary state while signaling link up changes.

IP_IDLE (Interconnection Port Idle state)

This state shall be reached if the interconnection port has a link and its port state is set to FORWARDING.

Local variables of the MIC protocol machine are listed in Table 52.

Table 52 – MRP Local variables of MIC protocol machine

Name	Type	Meaning
RPort_1	Unsigned16	Port identification of ring port 1
RPort_2	Unsigned16	Port identification of ring port 2
IPort	Unsigned16	Port identification of interconnection port
SA_Port1	OctetString[6]	Ring port RPort_1 MAC source address
SA_Port2	OctetString[6]	Ring port RPort_2 MAC source address
SA_IPort	OctetString[6]	Interconnection port IPort MAC source address
PRIORITY	Unsigned8	Priority according to IEEE 802.1Q for MRP-PDU. Shall be set to ORG.
MRP_TS_SA	OctetString[6]	MAC source address of host
IID	Unsigned16	Identification of interconnection domain
MRP_InLNKNReturn	Unsigned16	Counter, Range MRP_IN_LNKNRmax to 0
MRP_LNK_UP	Unsigned16	Constant value to indicate Link Up
MRP_LNK_DOWN	Unsigned16	Constant value to indicate Link Down

The MIC state machine for LC-mode shall be according to Table 53.

Table 53 – MIC State machine for LC-mode

#	Current state	Event /Condition =>action	Next state
1	PowerOn	=> InitFDB() AddMACFDB({RPort_1,RPort_2}, {MC_INCONTROL},ORG) AddMACFDB({local},{MC_INCONTROL},ORG) SetPortStateReq (IPort, BLOCKED) InterconnUpTimer.ini InterconnDownTimer.ini IID:= interconnection identifier	AC_STAT1
2	AC_STAT1	InterconnDownTimer expired /MPR_InLNKNReturn == 0 => MPR_InLNKNReturn:= MPR_IN_LNKNRmax	AC_STAT1
3	AC_STAT1	InterconnDownTimer expired /MPR_InLNKNReturn > 0 => MPR_InLNKNReturn:= MPR_InLNKNReturn – 1 InterconnDownTimer.start(MPR_IN_LNKdownT) InterconnLinkChangeReq (MPR_LNK_DOWN, MPR_InLNKNReturn X MPR_IN_LNKdownT)	AC_STAT1
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MPR_LNK_UP => MPR_InLNKNReturn:= MPR_IN_LNKNRmax InterconnDownTimer.stop InterconnUpTimer.start(MPR_IN_LNKupT) InterconnLinkChangeReq (MPR_LNK_UP, MPR_InLNKNReturn X MPR_IN_LNKupT)	PT
5	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /Link_status == MPR_LNK_DOWN => Ignore	AC_STAT1
6	AC_STAT1	InterconnTopologyChangeInd(MPR_SA, t, InID) /InID == IID => InterconnDownTimer.stop	AC_STAT1
7	AC_STAT1	InterconnLinkStatusInd (InID, RPort) /InID == IID => InterconnLinkChangeReq (MPR_LNK_DOWN, 0)	AC_STAT1

#	Current state	Event /Condition =>action	Next state
8	PT	InterconnUpTimer expired /MRP_InLNKNReturn == 0 => MRP_InLNKNReturn:= MRP_IN_LNKNRmax SetPortStateReq (IPort, FORWARDING)	IP_IDLE
9	PT	InterconnUpTimer expired /MRP_InLNKNReturn > 0 => MRP_InLNKNReturn:= MRP_InLNKNReturn – 1 InterconnUpTimer.start(MRP_LNKupT) InterconnLinkChangeReq (MRP_LNK_UP, MRP_InLNKNReturn X MRP_IN_LNKupT)	PT
10	PT	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnUpTimer.stop SetPortStateReq (IPort, BLOCKED) InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1
11	PT	MAUTypeChangeInd (RPort, Link_status) /Link_status == MRP_LNK_UP => Ignore	PT
12	PT	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnUpTimer.stop SetPortStateReq (IPort, FORWARDING)	IP_IDLE
13	PT	InterconnLinkStatusInd (InID, RPort) /InID == IID => InterconnLinkChangeReq (MRP_LNK_UP, 0)	PT
14	IP_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => MRP_InLNKNReturn:= MRP_IN_LNKNRmax SetPortStateReq (IPort, BLOCKED) InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1

#	Current state	Event /Condition =>action	Next state
15	IP_IDLE	MAUTypeChangeInd (RPort, Link_status) && Link_status == MRP_LNK_UP => Ignore	IP_IDLE
16	IP_IDLE	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => Ignore	IP_IDLE
17	IP_IDLE	InterconnLinkStatusInd (InID, RPort) /InID == IID => InterconnLinkChangeReq (MRP_LNK_UP, 0) InterconnForwardReq(IPort)	IP_IDLE
18	IP_IDLE	InterconnLinkChangeInd(InID, LinkStatus, RPort) /RPort != IPort && InID == IID => InterconnForwardReq(IPort)	IP_IDLE

The MIC state machine for RC-mode shall be according to Table 54

Table 54 – MIC State machine for RC-mode

#	Current state	Event /Condition =>Action	Next state
1	PowerOn	=> InitFDB() AddMACFDB({RPort_1,RPort_2,IPort}, {MC_INTEST},ORG) AddMACFDB({RPort_1,RPort_2}, {MC_INCONTROL},ORG) AddMACFDB({local},{MC_INCONTROL},ORG) SetPortStateReq (IPort, BLOCKED) InterconnUpTimer.ini InterconnDownTimer.ini IID:= interconnection identifier	AC_STAT1
2	AC_STAT1	InterconnDownTimer expired /MRP_InLNKNReturn == 0 => MRP_InLNKNReturn:= MRP_IN_LNKNRmax	AC_STAT1
3	AC_STAT1	InterconnDownTimer expired /MRP_InLNKNReturn > 0 => MRP_InLNKNReturn:= MRP_InLNKNReturn – 1 InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnDownTimer.stop InterconnUpTimer.start(MRP_IN_LNKupT) InterconnLinkChangeReq (MRP_LNK_UP, MRP_InLNKNReturn X MRP_IN_LNKupT)	PT
5	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /Link_status == MRP_LNK_DOWN => Ignore	AC_STAT1
6	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => InterconnDownTimer.stop	AC_STAT1
7	PT	InterconnUpTimer expired /MRP_InLNKNReturn == 0 => MRP_InLNKNReturn:= MRP_IN_LNKNRmax SetPortStateReq (IPort, FORWARDING)	IP_IDLE

#	Current state	Event /Condition =>Action	Next state
8	PT	InterconnUpTimer expired /MRP_InLNKNReturn > 0 => MRP_InLNKNReturn:= MRP_InLNKNReturn – 1 InterconnUpTimer.start(MRP_LNKupT) InterconnLinkChangeReq (MRP_LNK_UP, MRP_InLNKNReturn X MRP_IN_LNKupT)	PT
9	PT	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnUpTimer.stop SetPortStateReq (IPort, BLOCKED) InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1
10	PT	MAUTypeChangeInd (RPort, Link_status) /Link_status == MRP_LNK_UP => Ignore	PT
11	PT	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnUpTimer.stop SetPortStateReq (IPort, FORWARDING)	IP_IDLE
12	IP_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => MRP_InLNKNReturn:= MRP_IN_LNKNRmax SetPortStateReq (IPort, BLOCKED) InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1
13	IP_IDLE	MAUTypeChangeInd (RPort, Link_status) && Link_status == MRP_LNK_UP => Ignore	IP_IDLE
14	IP_IDLE	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => Ignore	IP_IDLE

#	Current state	Event /Condition =>Action	Next state
15	IP_IDLE	InterconnLinkChangeInd(InID, LinkStatus, RPort) /RPort != IPort && InID == IID => InterconnForwardReq(IPort)	IP_IDLE

8.2.9 MIM and MIC functions

The MIM and MIC functions shall be according to Table 55.

Table 55 – MRP Interconnection functions

Function name	Operations
InterconnTestReq (time)	SetupInterconnTestReq() InterconnTestTimer.start(time)
SetupInterconnTestReq()	Create MRP-PDU according MRP_InTest Assignments: MRP_Type:= MRP_InTest MRP_InID:= IID MRP_SA:= MRP_TS_SA MRP_PortRole:= frame sent on primary ring port or secondary ring port or interconnection port MRP_InState:= actual interconnection state MRP_Transition:= actual number of transitions between interconnection open state and interconnection closed state MRP_TimeStamp:= actual local counter value MRP_Type:= MRP_Common MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (IPort, MC_INTEST, SA_IPort, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_1, MC_INTEST, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_INTEST, SA_Port2, PRIORITY, LT, MRP-PDU)
InterconnTestInd (MRP_SA, RPort, InID)	Receive MRP-PDU according MRP_InTest MRP_SA:= MRP_SA from MRP-PDU RPort:= port which received this packet InID:= MRP_InID from MRP-PDU
InterconnForwardReq(RPort)	Function to send the received MRP-PDU at port RPort.
InterconnTopologyChangeReq(time)	SetupInterconnTopologyChangeReq(MRP_IN_TOPNRmax X time) if time == 0 ClearLocalFDB() else InterconnTopTimer.start(MRP_IN_TOPchgT)

Function name	Operations
SetupInterconnTopologyChangeReq(time)	Create MRP-PDU according MRP_InTopologyChange Assignments: MRP_Type:= MRP_InTopologyChange MRP_InID:= IID MRP_SA:= MRP_TS_SA MRP_Interval:= time MRP_Type:= MRP_Common MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_INCONTROL, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_INCONTROL, SA_Port2, PRIORITY, LT, MRP-PDU) SendFrameReq (IPort, MC_INCONTROL, SA_IPort, PRIORITY, LT, MRP-PDU)
InterconnTopologyChangeInd(MRP_SA, time, InID)	Receive MRP-PDU according MRP_InTopologyChange MRP_SA:= MRP_SA from MRP-PDU time:= MRP_Interval from MRP-PDU InID:= MRP_InID from MRP-PDU
InterconnLinkChangeReq(LinkStatus, time)	Create MRP-PDU according MRP_InLinkUp or MRP_InLinkDown Assignments: if LinkStatus == MRP_LNK_UP MRP_Type:= MRP_LinkUp else MRP_Type:= MRP_LinkDown MRP_InID:= IID MRP_SA:= MRP_TS_SA MRP_PortRole:= frame sent on primary ring port or secondary ring port or interconnection port MRP_Interval:= time MRP_Type:= MRP_Common MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_INCONTROL, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_INCONTROL, SA_Port2, PRIORITY, LT, MRP-PDU) SendFrameReq (IPort, MC_INCONTROL, SA_IPort, PRIORITY, LT, MRP-PDU)

Function name	Operations
InterconnLinkChangeInd(InID, LinkStatus, RPort)	Receive MRP-PDU according MRP_InLinkDown or MRP_InLinkUp InID:= MRP_InID from MRP-PDU RPort:= port which received this packet if MRP_Type == MRP_InLinkUp LinkStatus:= MRP_LNK_UP else LinkStatus:= MRP_LNK_DOWN
InterconnLinkStatusPollReq(time)	SetupInterconnLinkStatusPollReq() If time > 0 InterconnLinkStatusTimer.start(MRP_IN_LNKSTATchgT)
SetupInterconnLinkStatusPollReq()	Create MRP-PDU according MRP_In_LinkStatusPollRequest Assignments: InID:= IID MRP_SA:= MRP_TS_SA MRP_PortRole:= frame sent on primary ring port or secondary ring port or interconnection port MRP_Type:= MRP_Common MRP_SequenceID:= next SequenceID MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_INCONTROL, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_INCONTROL, SA_Port2, PRIORITY, LT, MRP-PDU)
InterconnLinkStatusInd(InID, RPort)	Receive MRP-PDU according MRP_InLinkStatusPoll InID:= MRP_InID from MRP-PDU RPort:= port which received this packet
MAUTypeChangeInd(RPort, Link_status)	Receive a local link change indication. RPort:= port which caused the local link change indication. Link_status:= MRP_LNK_UP or MRP_LNK_DOWN (depends on the the local link change indication)
SetPortStateReq(RPort, Status)	Function to set the port status of RPort to Status
ClearFDB(time)	FDBClearTimer.start(time)
ClearLocalFDB()	Function to clear the FDB within the MRP node. The learning of source addresses from ingress frames, which were sent out before the topology change was indicated, shall be prevented
InitFDB()	Function to initialize Filtering Database
AddMACFDB(Destination, MAC-Address, Priority)	Function to add Static Filtering Entries (MAC-Address) in the FDB with Priority and Destination. The term local in the state diagram means a connection to the Higher-Layer Entity (see IEEE 802.1D).
SendFrameReq(RPort, DestinationAddress, SourceAddress, Priority, LT, MRP-PDU)	Function to send an MRP-PDU at port RPort with the SourceAddress and LT to the DestinationAddress. Priority used in the TagControlInformation is coded in the frame if VLAN is used

8.2.10 Interconnection Topology Change timer

Interconnection topology change timer is an auxiliary state machine. Interconnection topology change timer shall be according to the state machine in Table 56.

Table 56 – MRP Interconnection topology change timer

#	Current state	Event /Condition =>Action	Next state
1	PowerOn	=> InterconnTopTimer.ini IN_TC_NReturn:= MRP_IN_TOPNRmax – 1	IDLE
2	IDLE	InterconnTopTimer expired /IN_TC_NReturn > 0 => SetupInterconnTopologyChangeReq (IN_TC_NReturn X MRP_IN_TOPchgT) IN_TC_NReturn:= IN_TC_NReturn – 1 InterconnTopTimer.start(MRP_IN_TOPchgT)	IDLE
3	IDLE	InterconnTopTimer expired /IN_TC_NReturn == 0 => IN_TC_NReturn:= MRP_IN_TOPNRmax – 1 ClearFDB(0) SetupInterconnTopologyChangeReq (0)	IDLE

8.2.11 Interconnection Link Status Poll timer

Interconnection link status poll timer is an auxiliary state machine. Interconnection link status poll timer shall be according to the state machine in Table 57.

Table 57 – MRP Interconnection link status poll timer

#	Current state	Event /Condition =>Action	Next state
1	PowerOn	=> InterconnLinkStatusTimer.ini MRP_IN_LNKSTAT_NReturn:= MRP_IN_LNKSTATNRmax – 1	IDLE
2	IDLE	InterconnLinkStatusTimer expired /MRP_IN_LNKSTAT_NReturn > 0 => SetupInterconnLinkStatusPollReq () MRP_IN_LNKSTAT_NReturn:= MRP_IN_LNKSTAT_NReturn – 1 InterconnLinkStatusTimer.start(MRP_IN_LNKSTATchgT)	IDLE
3	IDLE	InterconnLinkStatusTimer expired /MRP_IN_LNKSTAT_NReturn == 0 => MRP_IN_LNKSTAT_NReturn:= MRP_IN_LNKSTATNRmax – 1 SetupInterconnLinkStatusPollReq ()	IDLE

9 MRP installation, configuration and repair

9.1 Ring port and Interconnection port parameters

Ring port parameterization for the MRM and all MRC in a ring shall comply with the settings from Table 58.

MRP Interconnection port parameterization for the MIM and all MIC in an interconnection shall comply with the settings from Table 58.

Table 58 – MRP Network/Connection parameters

Parameter	Value
Link speed	The link speed shall be at least 100 Mbit/s
Duplex setting	Ring ports and the Interconnection port shall operate in full duplex mode, Administrative mode of a port may be set to autonegotiation, but negotiated value (oper mode) shall be full duplex

9.2 Ring topology parameters

The number of nodes participating in a ring shall not exceed 50.

NOTE 1 For more than this number of nodes in a ring, the maximum recovery time can be exceeded and the ring can become instable.

NOTE 2 For 10 ms maximum recovery time, the number of nodes can be lower than the above mentioned (see 9.9 for details).

9.3 MRM parameters

The MRM defines with its parameter set the maximum recovery time of a ring. Table 59 specifies four consistent sets of parameters for a maximum ring recovery time of 500 ms, 200 ms, 30 ms and 10 ms.

NOTE Additional consistent parameter sets for shorter or longer maximum recovery times can be supported in an MRM and MRC. The designer is responsible for the consistency for all parameters in Clause 9 (see the examples shown in 9.9). The installer is responsible for the consistency of all nodes in the ring.

Table 59 – MRP MRM parameters

Parameter	Max. recovery time				Meaning
	500 ms	200 ms	30 ms ^b	10 ms ^b	
MRP_TOPchgT	20 ms	10 ms	0.5 ms	0.5 ms	Topology Change (Clear Address Table) request interval
MRP_TOPNRmax	3	3	3	3	Topology Change (Clear Address Table) repeat count
MRP_TSTshortT	30 ms	10 ms	1 ms ^c	0.5 ms ^c	MRP_Test short interval
MRP_TSTdefaultT	50 ms	20 ms	3.5 ms ^c	1 ms ^c	MRP_Test default interval
MRP_TSTNRmax	5	3	3	3	MRP_Test monitoring count
MRP_TSTExtNRmax	15	N.A. ^a	N.A. ^a	N.A. ^a	MRP_Test extended monitoring count (option)
^a The following are required: The option "Non-blocking MRC supported" shall be set to FALSE for max. recovery time of 200 ms, 30 ms and 10 ms. All MRC shall support blocking mode. MRP_TSTExtNRmax not applicable. ^b The 10 ms and 30 ms recovery time puts additional restrictions on the cable length and on the reaction time of each client in the network (see 9.4 and 9.9 for details). ^c To reduce the load on network devices in case of "RING_OPEN" error (e.g. due to poor configuration), the intervals for MRP_TSTshortT and MRP_TSTdefaultT may be configured to take higher values in this case.					

9.4 MRC parameters and constraints

Table 60 specifies the MRC parameter sets (used for all four MRM parameter sets).

Table 60 – MRP MRC parameters

Parameter	Max. recovery time 500 ms and 200 ms	Max. recovery time 30 ms and 10 ms	Meaning
MRP_LNKdownT	20 ms	1 ms	Link Down Timer interval
MRP_LNKupT	20 ms	1 ms	Link Up Timer interval
MRP_LNKNRmax	4	4	Link Change (Up or Down) count

NOTE 1 These parameters are computed under the assumption that the traffic load in the ring does not exceed 90 %.

For the 10 ms and 30 ms recovery time, the MRC shall guarantee:

- fast flush of its FDB, see 9.9 for details;
- port state transition (e.g. from BLOCKED to NONBLOCKED): < 0,5 ms
- design for a short reaction time of a switch to an event to support quick reconfiguration.

NOTE 2 The measuring of some of these parameters can be done according to IEC 62439-1:2010 and IEC 62439-1:2010/AMD1:2012, Clause 8.

9.5 MRA compatibility to earlier Automanager protocol version

In addition to the MRA functionality described in this document, a user of MRA capable devices may claim support for the earlier Automanager protocol version for backward compatibility reasons.

The migration path to the standard Automanager function may be one of the following:

- the ring topology does not use the Automanager capabilities and all devices shall be configured manually to the role of either MRM or MRC. Only one device shall be configured to the role of MRM, the other devices shall be configured to the role of MRC;
- the ring topology uses the Automanager functionality described in this standard. Legacy devices not supporting the Automanager as defined in this standard shall be manually configured to the role of MRC. Devices supporting the Automanager functionality as defined in this standard may participate in the voting process and vote the MRM;
- the ring topology uses the earlier Automanager protocol which is not fully compliant with the Automanager functionality described in this standard. In this case it is necessary for migration purposes, that devices according to this standard support the additional MRP_Option and MRP_Suboption fields handling as described in Annex A. This solution is a valid migration path to the Automanager function introduced in this document and thus protects the installed base of field devices.

9.6 Interconnection topology parameters

The interconnection links between two MRP rings shall not contain further nodes.

NOTE The connected rings become instable if an MRC in an interconnection link detects a failure of a ring port and notifies this change by sending an MRP_LinkDown frame to the ring.

9.7 MIM parameters

The MIM defines with its set of parameters the maximum recovery time of an MRP interconnection. Table 61 specifies two consistent sets of parameters for maximum MRP interconnection recovery times of 500 ms and 200 ms.

NOTE Additional consistent parameter sets for shorter or longer maximum recovery times can be supported in an MIM and MIC. The designer is responsible for the consistency for all parameter in Clause 9 (see the examples shown in 9.9). The installer is responsible for the consistency of all nodes in the ring and in the ring interconnection.

Table 61 – MRP MIM parameters

Parameter	Max. recovery time		Meaning
	500 ms	200 ms	
MRP_IN_TOPchgT	20 ms	10 ms	Interconnection Topology Change (Clear Address Table) request interval
MRP_IN_TOPNRmax	3	3	Interconnection Topology Change (Clear Address Table) repeat count
MRP_IN_TSTdefaultT	50 ms ^a	20 ms ^a	MRP_InTest default interval
MRP_IN_TSTNRmax	8	8	MRP_InTest monitoring count
MRP_IN_LNKSTATchgT	20 ms	20 ms	Interconnection Link Status Poll interval
MRP_IN_LNKSTATNRmax	8	8	MRP_InLinkStatusPoll monitoring count
^a To reduce the load on network devices in case of "INTERCONNECTION_OPEN" (e.g. due to poor configuration), the interval for MRP_IN_TSTdefaultT may be configured to take higher values in this case.			

9.8 MIC parameters and constraints

Table 62 specifies the MIC parameter sets (used for all MIM parameter sets).

Table 62 – MRP MIC parameters

Parameter	Max. recovery time 500 ms and 200 ms	Meaning
MRP_IN_LNKdownT	20 ms	Interconnection Link Down Timer interval
MRP_IN_LNKupT	20 ms	Interconnection Link Up Timer interval
MRP_IN_LNKNRmax	4	Interconnection Link Change (Up or Down) count
NOTE These parameters are computed under the assumption that the traffic load in the ring and in the ring interconnection does not exceed 90 %.		

NOTE The measuring of some of these parameters can be done according to IEC 62439-1:2010 and IEC 62439-1:2010/AMD1:2012, Clause 8.

9.9 Calculation of MRP ring recovery time

9.9.1 Overview

For a recovery time of 10 ms, the actual recovery time of the network also depends on the number of participating devices. The forwarding time of each device and the delay frames experience while travelling through the ring network needs to be accounted for. To calculate an actual figure for the reconfiguration time of a network, the following formulas described in 9.9.2 can be used. In 9.9.3, a calculation example is given for a worst case scenario. In 9.9.4, an example is given to achieve the best possible reconfiguration time for 50 devices with the consistent set of parameters for 30 ms.

NOTE Because of the high frequency of test frames for 10 ms and 30 ms recovery time, the likelihood of the detection of an error via the test frames is higher than the detection via Link Down messages. Therefore, a possible faster detection via Link Down messages is not the subject of the following calculations.

9.9.2 Deduction of formula

The actual ring recovery time is dependent on different variables, that all have to be factored in:

An MRM detects a failure of the ring after not receiving its test frames for a defined time period. This time T_{test} is described by the following formula:

$$T_{\text{test}} = \text{MRP_TSTdefaultT} \times \text{MRP_TSTNRmax}$$

where

MRP_TSTdefaultT and MRP_TSTNRmax are parameters according to Table 59.

The total time an MRM needs to detect a failure is not only T_{test} , but also the dead time from the time of the actual moment a failure in the network occurs till the last reception of a test frame at the MRM. The actual moment of failure can be anywhere between the passing of the last MRM test frame at the point of failure prior to failure and the expected time of arrival of the next test frame at the point of failure, which will not be transmitted anymore. Because the exact point of failure in the topology cannot be predetermined, the time an MRP test frame takes to travel through the whole ring, T_{ring} shall be taken as a basis to calculate this dead time:

$$T_{\text{ring}} = N \times (T_{\text{switch}} + T_{\text{queue}} + T_{\text{bit}} + T_{\text{line}})$$

where

N is the total number of participating ring devices;

T_{switch} is the delay introduced by each switching node;

T_{queue} is the delay introduced by a frame that is being transmitted to the network, thus delaying the transmission of the MRP test frame, even if the MRP test frame is handled in a high priority switch queue;

T_{bit} is the time a device needs to send and receive MRP test traffic (typically 5,12 μs at 100 Mbit/s Ethernet for a 64 octets frame);

T_{line} is the propagation delay of the frames on the physical medium. If the distance between the switching nodes is not exceptionally large, this can be neglected.

Consequently, the actual time from the occurrence of the failure to its detection in the MRM, T_{detect} , is calculated as follows:

$$T_{\text{detect}} = T_{\text{test}} + T_{\text{ring}} \text{ or}$$

$$T_{\text{detect}} = \text{MRP_TSTdefaultT} \times \text{MRP_TSTNRmax} + N \times (T_{\text{switch}} + T_{\text{queue}} + T_{\text{bit}} + T_{\text{line}}).$$

After the detection of the failure, the MRM starts to transmit topology change frames to initiate an FDB flush in the ring devices after a hold down time T_{hold} . This time needs to be added to the total reconfiguration time. In a worst case scenario, the time a topology change frame needs to reach all devices in the ring is also T_{ring} , because the MRP_TopologyChange frame, analogous to the MRP_Test frame, may have to travel through the whole ring. The time the last device needs to flush its FDB, T_{FDB} , needs to be added, because the ring is only operational after the last device has flushed its FDB, all other devices have flushed their FDBs sooner. So the time for the ring to reconfigure is extended by the following T_{flush} :

$$T_{\text{flush}} = T_{\text{hold}} + T_{\text{ring}} + T_{\text{FDB}}$$

where

$$T_{\text{hold}} = \text{MRP_TOPchgT} \times \text{MRP_TOPNRmax}, \text{ taken from Table 61;}$$

T_{FDB} is the time a device needs to flush its FDB.

For the total reconfiguration time of the ring T_{rec} , T_{detect} and T_{flush} need to be added:

$$T_{\text{rec}} = T_{\text{detect}} + T_{\text{flush}} \text{ or}$$

$$T_{\text{rec}} = \text{MRP_TSTdefaultT} \times \text{MRPTSTNRmax} + 2 \times N \times (T_{\text{switch}} + T_{\text{queue}} + T_{\text{bit}} + T_{\text{line}}) + T_{\text{FDB}} + T_{\text{hold}}$$

EXAMPLE Calculation with the consistent set of parameters for 10 ms recovery time:

$$\text{MRP_TSTdefaultT} = 1 \text{ ms}$$

$$\text{MRP_TSTNRmax} = 3$$

$$N = 50$$

$$T_{\text{switch}} = 10 \mu\text{s}$$

$T_{\text{queue}} = 0$, assuming that the MRP frames in the high priority queue are not delayed by any frames already in the buffer of a ring device and being transmitted on the network.

$$T_{\text{bit}} = 5,12 \mu\text{s}$$

$T_{\text{line}} = 0,5 \mu\text{s}$ for 100 m cable between every two network devices

$T_{\text{FDB}} = 500 \mu\text{s}$

$T_{\text{hold}} = 0,5 \text{ ms} \times 3$, according to the consistent set of parameters for 10 ms

With this:

$T_{\text{rec}} = 1 \text{ ms} \times 3 + 2 \times 50 \times (10 \mu\text{s} + 5,12 \mu\text{s} + 0,5 \mu\text{s}) + 0,5 \text{ ms} + 1,5 \text{ ms} \approx 6,6 \text{ ms}$

This is a calculation under best case assumptions. It is assumed that the MRP test frames do not experience delay through transmission of other frames and that MRP traffic is transmitted under the best possible circumstances.

9.9.3 Worst case calculation for recovery time of 10 ms

In a worst case scenario, each frame vital to the ring reconfiguration experiences delay in each device on the network, introduced by a large Ethernet frame the size of 1 522 octets, which is being forwarded prior to the forwarding of the MRP frame. The example calculation thus has to factor in $T_{\text{queue}} = 122 \mu\text{s}$ for each ring device. This introduces two timing constraints that both shall be observed at the same time:

- The time $T_{\text{ring}} + \text{MRP_TSTdefaultT}$ shall not exceed T_{test} . If it were, an open ring could be falsely detected in the MRM in a situation where the delay between two test frames immediately sent after each other is varying strongly, e.g. due to network congestion. This means that T_{ring} for 10 ms recovery time may not exceed 2 ms.
- The total reconfiguration time that can be achieved in the ring shall not exceed 10 ms.

EXAMPLE Calculation for the worst case scenario with the consistent set of parameters for 10 ms recovery time:

$\text{MRP_TSTNRmax} = 3$

$N = 14$

$T_{\text{switch}} = 10 \mu\text{s}$

$T_{\text{queue}} = 122 \mu\text{s}$, assuming that the MRP frames in the high priority queue are being delayed in each switching node by a frame of 1 522 octets already in the buffer and being transmitted on the network.

$T_{\text{queue}} = 5,12 \mu\text{s}$

$T_{\text{line}} = 0$, assuming the line delay is negligible.

$T_{\text{FDB}} = 500 \mu\text{s}$

$T_{\text{hold}} = 0,5 \text{ ms} \times 3 = 1,5 \text{ ms}$

With this:

$T_{\text{ring}} = 14 \times (10 \mu\text{s} + 122 \mu\text{s} + 5,12 \mu\text{s}) \approx 1,9 \text{ ms}$

$T_{\text{rec}} = 1 \text{ ms} \times 3 + 2 \times 14 \times (10 \mu\text{s} + 122 \mu\text{s} + 5,12 \mu\text{s}) + 0,5 \text{ ms} + 1,5 \text{ ms} \approx 8,8 \text{ ms}$

This shows that under the assumption of the timing parameters above and in the worst case scenario, the 10 ms recovery time can be achieved when the number of devices in the ring (N) is 14 or lower.

9.9.4 Worst case calculation for 50 devices

With N=50 devices, the consistent set of parameters for 30 ms recovery time and the same worst case scenario as described in 9.9.3, the following T_{ring} can be achieved:

$$T_{ring} = 50 \times (10 \mu s + 122 \mu s + 5,12 \mu s) \approx 6,9 \text{ ms}$$

When $MRP_TSTdefaultT$ is extended to 3,5 ms, $T_{ring} + MRP_TSTdefaultT \approx 10,4 \text{ ms}$ is below the threshold for open ring detection in the MRM, T_{test} , which is for the consistent set of parameters for 30 ms:

$$T_{test} = MRP_TSTdefaultT \times MRP_TSTNRmax = 3,5 \text{ ms} \times 3 = 10,5 \text{ ms}.$$

This results in the following total reconfiguration time for the ring for 50 devices in a worst case situation, with the consistent set of parameters for 30 ms:

$$T_{rec} = 3,5 \text{ ms} \times 3 + 2 \times 50 (10 \mu s + 122 \mu s + 5,12 \mu s) + 0,5 \text{ ms} + 1,5 \text{ ms} \approx 26,2 \text{ ms}$$

NOTE To provide a viable means of calculating the actual recovery time of a network, a device manufacturer claiming support for 10 ms and 30 ms recovery time can disclose T_{switch} and T_{FDB} for the supporting devices.

9.10 Calculation of MRP Automanager voting time

As outlined in 5.10.3, the manager voting is an iterative process. With each iteration, two out of three Automanagers change into client role. The number of iterations n_{it} required to vote the best manager depends on the priority order and the number of devices.

Each iteration requires one MRP_Test cycle at a maximum. The time needed to elect the MRM is thus depending on the used MRM consistent parameter set as defined in Table 59.

Assuming for example a ring of 50 Automanagers, using the 200 ms MRM consistent parameter set, with a worst case priority distribution, the iteration would need $n_{it} = 4$ iteration steps and the manager voting would thereby need:

$$T_{vot} \approx MRP_TSTdefaultT \times n_{it} = 20 \text{ ms} \times 4 \approx 80 \text{ ms}$$

to elect the finally remaining MRM.

10 MRP Management Information Base (MIB)

10.1 General

For the configuration of MRP nodes there are two optional network Management Information Bases (MIBs).

Subclause 10.2 provides an MIB with a monitoring view. This can be used for field devices.

Subclause 10.3 provides an MIB with a management and monitoring view. This can be used for network components.

10.2 MRP MIB with a monitoring view

NOTE When copying the MIB from a PDF document, all headers and footers are removed and the characters are converted to the character format that SNMP requires. Although care has been taken not to use illegal characters, some can appear in the final document due to the editing process.

```

-- *****
-- Monitoring MIB definitions
-- *****

IEC-62439-2-MIB          DEFINITIONS:= BEGIN

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

IMPORTS
    MODULE-IDENTITY,
    OBJECT-IDENTITY,
    OBJECT-TYPE,
    TimeTicks,
    Counter32,
    Unsigned32,
    Integer32                FROM SNMPv2-SMI
    OBJECT-GROUP,
    MODULE-COMPLIANCE        FROM SNMPv2-CONF
    TEXTUAL-CONVENTION,
    DisplayString            FROM SNMPv2-TC;
-- *****
-- Root OID
-- *****
iec62439  MODULE-IDENTITY
    LAST-UPDATED "201405090000Z" -- May 09, 2014
    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.
        This definitions specify the pure monitoring variant
        of a SNMP entity. If the SNMP entity also contains
        management functionality, then IEC 62439 management MIB
        definitions shall be used instead of this MIB module.
    "

    REVISION      "201405090000Z" -- May 09, 2014
    DESCRIPTION "
        Manager (auto) support has been added in mrpDomainAdminRole.
        Interconnection functionality added.
        Units of mrpDomainRoundTripDelayMax and mrpDomainRoundTripDelayMin
        have been corrected to milliseconds.
        mrpDomainOperRole and mrpDomainAdminRole data type have been set
        to INTEGER.
        Missing constructs in import area have been added.
        Module identity block has been put on top.
    "

    REVISION      "200811100000Z" -- November 10, 2008
    DESCRIPTION "
        The undefined use of the value 'badValue' for objects
        of the optional 'mrpDomainDiagGroup' has been removed
        from DESCRIPTION clause at the objects concerned.
        The role dependency of the objects 'mrpDomainState' and
        'mrpDomainError' in the DESCRIPTION clause removed.
    "

    REVISION      "200811100000Z" -- November 10, 2008
    DESCRIPTION "
        Separation of IEC 62439 into a suite of documents
        This MIB applies to IEC 62439-2, no change in functionality
    "

    REVISION      "200708240000Z" -- August 24, 2007

```

```

DESCRIPTION "
    Final Draft Proposal version of the Network Management
    interface for the Media Redundancy Protocol (MRP) with
    the division in pure Monitoring functionality and in
    additional management functionality.
"

 ::= { iso std(0) 62439 }

-- *****
-- Declaration of IEC62439UuidType
-- *****
IEC62439UuidType ::= TEXTUAL-CONVENTION
    STATUS         current
    DESCRIPTION "
        The IEC 61158-5-10 defines the structure of
        the UUID as a data type numeric
        identifier 1025.
    "
    SYNTAX         OCTET STRING (SIZE (16))

-- *****
-- Redundancy Protocols
-- *****
mrp          OBJECT IDENTIFIER ::= { iec62439 1 }
prp          OBJECT IDENTIFIER ::= { iec62439 2 }
crp          OBJECT IDENTIFIER ::= { iec62439 3 }
brp          OBJECT IDENTIFIER ::= { iec62439 4 }

-- *****
-- objects groups of redundancy object identifiers
-- *****
mrpObjects   OBJECT IDENTIFIER ::= { mrp 1 }
mrpNotifications OBJECT IDENTIFIER ::= { mrp 2 }
mrpConformance OBJECT IDENTIFIER ::= { mrp 3 }

-- *****
-- Objects of the MRP Network Management
-- *****
mrpDomainTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MrpDomainEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "
        Every entry in this table contains information about
        a Media Redundancy Protocol (MRP) domain.

        Entries cannot be created or deleted via SNMP
        operations.
    "
 ::= { mrpObjects 1 }

mrpDomainEntry OBJECT-TYPE
    SYNTAX      MrpDomainEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the mrpDomainTable."
    INDEX       { mrpDomainIndex }
 ::= { mrpDomainTable 1 }

MrpDomainEntry ::= SEQUENCE {
    mrpDomainIndex      Unsigned32,
    mrpDomainID         IEC62439UuidType,
    mrpDomainName       DisplayString,
    mrpDomainAdminRole  INTEGER,
    mrpDomainOperRole   INTEGER,
    mrpDomainRingPort1  Integer32,
    mrpDomainRingPort1State INTEGER,
    mrpDomainRingPort2  Integer32,
    mrpDomainRingPort2State INTEGER,
    mrpDomainState      BITS,
    mrpDomainError       BITS,
    mrpDomainBlocked     INTEGER,
    mrpDomainVlanId      Unsigned32,
    mrpDomainManagerPriority INTEGER,
    mrpDomainRingOpenCount Counter32,
    mrpDomainLastRingOpenChange TimeTicks,
    mrpDomainRoundTripDelayMax Unsigned32,

```

```

        mrpDomainRoundTripDelayMin      Unsigned32,
        mrpDomainResetRoundTripDelays   INTEGER,
        mrpDomainMRMReactOnLinkChange   INTEGER
    }

mrpDomainIndex      OBJECT-TYPE
    SYNTAX            Unsigned32
    MAX-ACCESS        not-accessible
    STATUS             current
    DESCRIPTION        "The index of the entry"
    ::= { mrpDomainEntry 1 }

mrpDomainID         OBJECT-TYPE
    SYNTAX            IEC62439UuidType
    MAX-ACCESS        read-only
    STATUS             current
    DESCRIPTION        "
        Universally unique identifier belongs to the MRP domain
        which represents a ring.
    "
    ::= { mrpDomainEntry 2 }

mrpDomainName       OBJECT-TYPE
    SYNTAX            DisplayString
    MAX-ACCESS        read-only
    STATUS             current
    DESCRIPTION        "
        A logical name for the MRP domain to ease the
        management of MRP domains.
    "
    ::= { mrpDomainEntry 3 }

mrpDomainAdminRole  OBJECT-TYPE
    SYNTAX            INTEGER {
        disable(0),
        client(1),
        manager(2),
        managerAutoComp(3),
        managerAuto(4)
    }
    MAX-ACCESS        read-write
    STATUS             current
    DESCRIPTION        "
        Control the MRP behavior of the system per domain.

        If the value is set to disable(0) the MRP entity of
        this domain shall be disabled.
        If the value is set to client(1) the entity shall be set
        to the role of a Media Redundancy Client (MRC).
        If the value is set to manager(2) the entity shall be
        set to the role of a Media Redundancy Manager (MRM).
        If the value is set to managerAutoComp(3) the entity shall be
        set to the role of a Media Redundancy Manager Auto (MRA)
        complying to Annex A.
        If the value is set to managerAuto(4) the entity shall be
        set to the role of a Media Redundancy Manager Auto (MRA) not
        supporting Annex A.
        The factory settings are recommended to adjust the
        value of this object to the client(1) capability of the
        component, or, if supported, to the managerAutoComp(3)
        or managerAuto(4) capability,
        in order to prevent multiple managers are
        in ring (the order of the capabilities are not
        necessarily conform to the order of the object values
        here).
        If the agent restricts the write access, no matter what
        reason, it shall reject write requests by responding
        with 'badValue'.
    "
    ::= { mrpDomainEntry 4 }

mrpDomainOperRole   OBJECT-TYPE
    SYNTAX            INTEGER {
        disable(0),
        client(1),
        manager(2),
        managerAutoComp(3),
        managerAuto(4)
    }

```

```

    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "
                    The operational role of an MRP entity per domain.
                    A value of disabled(0) signals that the entity doesn't
                    work (whatever reason).
                    A value of client(1) signals that the entity is in a
                    client role.
                    A value of manager(2) signals that the entity is the
                    manager of this MRP domain.
                    A value of managerAutoComp(3) signals that the entity is
                    in automanager role complying to Annex A.
                    A value of managerAuto(4) signals that the entity is in
                    automanager role not supporting Annex A.
                    "
 ::= { mrpDomainEntry 5 }

mrpDomainRingPort1 OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "
                    The ifIndex of the layer 2 interface which is used
                    as ring port 1.
                    "
 ::= { mrpDomainEntry 6 }

mrpDomainRingPort1State OBJECT-TYPE
    SYNTAX          INTEGER {
                        disabled(1),
                        blocked(2),
                        forwarding(3),
                        not-connected(4)
                    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "
                    Operational state of the first Ring-Port.

                    disabled(1) indicates that all frames are dropped.
                    blocked(2) indicates that all frames are dropped except
                        – MRP topology change frames and MRP test frames from
                          a MRM,
                        – MRP link change frames from an MRC,
                        – MRP interconnection topology change from a MIM,
                        – MRP interconnection link change from a MIC,
                        – frames from other protocols that also define to pass
                          blocked(2) ports.
                    forwarding(3) indicates that all frames are passed through
                        according to the forwarding behavior
                        of IEEE 802.1D.
                    not-connected(4) indicates that the port has no link.
                    "
 ::= { mrpDomainEntry 7 }

mrpDomainRingPort2 OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "
                    The ifIndex of the layer 2 interface which is
                    used as ring port 2.
                    "
 ::= { mrpDomainEntry 8 }

mrpDomainRingPort2State OBJECT-TYPE
    SYNTAX          INTEGER {
                        disabled(1),
                        blocked(2),
                        forwarding(3),
                        not-connected(4)
                    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "
                    Operational state of the second Ring-Port.

```

```

disabled(1)      indicates that all frames are dropped.
blocked(2)      indicates that all frames are dropped except
                -MRP topology change frames and MRP test frames from
                  a MRM,
                -MRP link change frames from an MRC,
                -MRP interconnection topology change from a MIM,
                -MRP interconnection link change from a MIC,
                -frames from other protocols that also define to pass
                  blocked(2) ports.
forwarding(3)    indicates that all frames are passed through
                according to the forwarding behavior
                of IEEE 802.1D.
not-connected(4) indicates that the port has no link.
"
 ::= { mrpDomainEntry 9 }

mrpDomainState  OBJECT-TYPE
SYNTAX          BITS {
                disabled(0),
                undefined(1),
                ringOpen(2),
                reserved(3)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "
                Operational status of the MRP entity.
                disabled(0) -MRP switched off. All higher bits are invalid
                           and shall be reset.
                undefined(1) -Value is not valid.
                           All higher bits are invalid and shall be reset.
                ringOpen(2) -MRP ring redundancy lost
                           All higher bits are invalid and shall be reset.
                reserved(3) -reserved for further extensions.
                "
 ::= { mrpDomainEntry 10 }

mrpDomainError  OBJECT-TYPE
SYNTAX          BITS {
                noError(0),
                invalidVlanId(1),
                invalid(2),
                multipleMRM(3),
                singleSideReceive(4)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "
                If the device couldn't switch to the desired administrative state
                (thus the value of 'mrpDomainState' is not the expected one),
                this value provides the reason. Otherwise the bit noError(0) is set.

                noError(0)      -the operational state of the device is conform
                           to administrative state.
                           All higher bits are invalid and shall be reset.
                invalidVlanId(1) -the assigned VLAN ID is not permitted .
                invalid (2)     -Value is not valid.
                           All higher bits are invalid and shall be reset.
                multipleMRM(3)  -multiple active managers in ring domain.
                singleSideReceive(4) -the test frames of an MRM
                           have been seen, but only on one port.
                "
 ::= { mrpDomainEntry 11 }

mrpDomainBlocked OBJECT-TYPE
SYNTAX          INTEGER {
                enabled(1),
                disabled(2)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "
                The entity supports blocked ring ports.
                Shows whether a mrp domain requires the support of the
                BLOCKED port state at ring ports.
                The manager of a mrp domain decides whether this support

```

```

        is required.
        Set to enabled the manager demands that all clients
        shall support the blocked attribute also set to enabled.
        If mrpDomainBlocked is set disabled at the manager,
        then the value of mrpDomainBlocked can be arbitrary
        at the clients.
    enabled(1) client: supports ring ports whose port state can be
                  blocked.
                manager: works only with clients supporting blocked
                  ring ports.
    disabled(2) client: no support of blocked ring ports.
                manager: Work with clients supporting blocked ring
                  ports and with clients not supporting
                  blocked ring ports.
    "
 ::= { mrpDomainEntry 12 }

mrpDomainVlanId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The VLAN ID assigned to the MRP protocol.
        The VLAN ID only is in use when the bit invalidVlanId
        is not set in mrpDomainError.
        If value is set to 0 no VLAN is assigned.
        The invalidVlanId shall be set to 0 (no Error).
    "
 ::= { mrpDomainEntry 13 }

-- *****
-- Objects of the MRP Network Management for manager role
-- *****
mrpDomainManagerPriority OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The priority of this MRP entity. If the device is
        client only, the value of this object shall be ignored
        by the MRP entity. Only the four most significant bits
        shall be used, the bits 0 to 11 are reserved. The
        smaller value has the higher priority.
    "
 ::= { mrpDomainEntry 14 }

mrpDomainRingOpenCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Counter for ring-state changes to open.
    "
 ::= { mrpDomainEntry 15 }

mrpDomainLastRingOpenChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Timeticks since last change of ring-state
        to ring open.
    "
 ::= { mrpDomainEntry 16 }

mrpDomainRoundTripDelayMax OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The max. Round-Trip-Delay (in milliseconds)
        which was measured since startup.
    "
 ::= { mrpDomainEntry 17 }

mrpDomainRoundTripDelayMin OBJECT-TYPE
    SYNTAX      Unsigned32

```

```

    UNITS                "milliseconds"
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION            "
                            The min. Round-Trip-Delay (in milliseconds)
                            which was measured since startup.
                            "

 ::= { mrpDomainEntry 18 }

mrpDomainResetRoundTripDelays OBJECT-TYPE
    SYNTAX                INTEGER {
                            resetDelays(1)
                        }
    MAX-ACCESS            read-write
    STATUS                current
    DESCRIPTION            "
                            A write request with resetDelays(1) shall reset
                            the values of mrpDomainRoundTripDelayMax and
                            mrpDomainRoundTripDelayMin to zero at the
                            same time.
                            "

 ::= { mrpDomainEntry 19 }

mrpDomainMRMReactOnLinkChange OBJECT-TYPE
    SYNTAX                INTEGER {
                            enabled(1),
                            disabled(2)
                        }
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION            "
                            Tells whether the MRM reacts on link down MRP link change
                            frames.
                            enabled(1) indicates that the MRM reacts immediately on link down MRP
                            link change frames.
                            disabled(2) indicates that the MRM does not react on
                            link down MRP link change frames.
                            "

 ::= { mrpDomainEntry 20 }

-- *****
-- Objects of the Media redundancy Interconnection
-- *****
mrpInterconnectionTable OBJECT-TYPE
    SYNTAX                SEQUENCE OF MrpInterconnectionEntry
    MAX-ACCESS            not-accessible
    STATUS                current
    DESCRIPTION            "
                            Every entry in this table contains information about
                            a Media redundancy Interconnection domain.

                            Entries cannot be created or deleted via SNMP
                            operations.
                            "

 ::= { mrpObjects 2 }

mrpInterconnectionEntry OBJECT-TYPE
    SYNTAX                MrpInterconnectionEntry
    MAX-ACCESS            not-accessible
    STATUS                current
    DESCRIPTION            "An entry in the mrpInterconnectionTable."
    INDEX                { mrpInterconnectionID }
 ::= { mrpInterconnectionTable 1 }

MrpInterconnectionEntry ::= SEQUENCE {
    mrpInterconnectionID                Unsigned32,
    mrpInterconnectionName              DisplayString,
    mrpInterconnectionPort              Integer32,
    mrpInterconnectionPortState         INTEGER,
    mrpInterconnectionVlanId            Unsigned32,
    mrpInterconnectionMode              INTEGER,
    mrpInterconnectionAdminRole         INTEGER,
    mrpInterconnectionOperRole          INTEGER,
    mrpInterconnectionState             INTEGER,
    mrpInterconnectionError             BITS,
    mrpInterconnectionOpenCount          Counter32,
    mrpInterconnectionLastOpenChange    TimeTicks,
    mrpInterconnectionPeerMAC           OCTET STRING,

```

```

        mrpInterconnectionOwnMRMMAC      OCTET STRING,
        mrpInterconnectionPeerMRMMAC     OCTET STRING,
        mrpInterconnectionPeerRingDomainUUID IEC62439UuidType
    }

mrpInterconnectionID      OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION  "The index of the entry"
    ::= { mrpInterconnectionEntry 1 }

mrpInterconnectionName    OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
        A logical name for the Media redundancy Interconnection domain to
        ease the management of Media redundancy Interconnection domains.
    "
    ::= { mrpInterconnectionEntry 2 }

mrpInterconnectionPort    OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
        The ifIndex of the layer 2 interface which is used
        as interconnection port.
    "
    ::= { mrpInterconnectionEntry 3 }

mrpInterconnectionPortState OBJECT-TYPE
    SYNTAX      INTEGER {
        disabled(1),
        blocked(2),
        forwarding(3),
        notConnected(4)
    }
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
        Operational state of the Interconnection port.

        disabled(1) indicates that all frames are dropped.
        blocked(2)  indicates that all frames are dropped except
            -MRP Interconnection topology change frames,
            -MRP Interconnection test frames,
            -MRP Interconnection link change frames,
            -MRP Interconnection link status poll frames,
            -frames from other protocols that also define to pass
              blocked(2) ports.
        forwarding(3) indicates that all frames are passed through
            according to the forwarding behavior of IEEE 802.1D.
        notConnected(4) indicates that the port has no link.
    "
    ::= { mrpInterconnectionEntry 4 }

mrpInterconnectionVlanId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
        The VLAN ID assigned to the MRP Interconnection protocol.
        The VLAN ID only is in use when mrpInterconnectionError
        does not signal invalidVlanId.
        If value is set to 0 no VLAN is assigned.
    "
    ::= { mrpInterconnectionEntry 5 }

mrpInterconnectionMode    OBJECT-TYPE
    SYNTAX      INTEGER {
        undefined(1),
        lcMode(2),
        rcMode(3)
    }
    MAX-ACCESS   read-write
    STATUS      current

```

```

DESCRIPTION      "
                  Control the MRP Interconnection mode.

                  The value undefined(1) signals that the interconnection mode is
                  neither set to lcMode(2) nor to rcMode(3).
                  If the value is set to lcMode(2) the device in Media redundancy
                  Interconnection Manager (MIM) role is using the link check mode to
                  detect open or closed interconnection links.
                  If the value is set to rcMode(3) the device in Media redundancy
                  Interconnection Manager (MIM) role is using the ring check mode to
                  detect open or closed interconnection links.

                  The factory settings are recommended to adjust the
                  value of this object to the lcMode(2) capability.
                  If the agent restricts the write access, no matter what
                  reason, it shall reject write requests by responding
                  with 'badValue'.
                  "

 ::= { mrpInterconnectionEntry 6 }

mrpInterconnectionAdminRole  OBJECT-TYPE
    SYNTAX      INTEGER {
                  disable(0),
                  client(1),
                  manager(2)
                }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "
                  Control the MRP Interconnection behavior of the system per
                  Media redundancy Interconnection domain.

                  If the value is set to disable(0) the MRP Interconnection
                  entity of this domain shall be disabled.
                  If the value is set to client(1) the entity shall be set
                  to the role of a Media redundancy Interconnection Client (MIC).
                  If the value is set to manager(2) the entity shall be set
                  to the role of a Media redundancy Interconnection Manager (MIM).
                  The factory settings are recommended to adjust the
                  value of this object to the client(1) capability of the
                  component in order to prevent multiple managers are
                  in the interconnection (the order of the capabilities are not
                  necessarily conform to the order of the object values
                  here).
                  If the agent restricts the write access, no matter what
                  reason, it shall reject write requests by responding
                  with 'badValue'.
                  "

 ::= { mrpInterconnectionEntry 7 }

mrpInterconnectionOperRole  OBJECT-TYPE
    SYNTAX      INTEGER {
                  disable(0),
                  client(1),
                  manager(2)
                }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
                  The operational role of an MRP Interconnection entity
                  per Media redundancy Interconnection domain.
                  A value of disabled(0) signals that the entity doesn't
                  work (whatever reason).
                  A value of client(1) signals that the entity is in a
                  Media redundancy Interconnection Client role.
                  A value of manager(2) signals that the entity is the
                  manager of this Media redundancy Interconnection domain.
                  "

 ::= { mrpInterconnectionEntry 8 }

mrpInterconnectionState  OBJECT-TYPE
    SYNTAX      INTEGER {
                  disable(0),
                  undefined(1),
                  interconnOpen(2),
                  interconnClosed(3)
                }
    MAX-ACCESS   read-only

```

```

STATUS          current
DESCRIPTION      "
                  Operational status of the MRP Interconnection entity.
                  disable(0)      - The entity is disabled.
                  undefined(1)    - Value is not valid.
                  interconnOpen(2) - MRP Interconnection redundancy lost of this Media
                                redundancy Interconnection domain.
                  interconnClosed(3) - MRP Interconnection redundancy available of this Media
                                redundancy Interconnection domain.
                  "
 ::= { mrpInterconnectionEntry 9 }

mrpInterconnectionError OBJECT-TYPE
    SYNTAX      BITS {
                  noError(0),
                  invalidVlanId(1)
                }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                  Configuration status of the MRP Interconnection entity.

                  noError(0)      - no configuration error detected.
                  invalidVlanId(1) - the assigned VLAN ID is not permitted.
                  "
 ::= { mrpInterconnectionEntry 10 }

mrpInterconnectionOpenCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                  Counter for MRP Interconnection state changes to
                  interconnOpen.
                  "
 ::= { mrpInterconnectionEntry 11 }

mrpInterconnectionLastOpenChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                  Timeticks since last change of
                  mrpInterconnectionState to interconnOpen.
                  "
 ::= { mrpInterconnectionEntry 12 }

mrpInterconnectionPeerMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                  Device-MAC-Address of the devices interconnection peer, connected
                  to the devices interlink port. This variable contains
                  the MAC-Address contained in the media link check frame received at the
                  interconnection port.
                  "
 ::= { mrpInterconnectionEntry 13 }

mrpInterconnectionOwnMRMMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                  The device-MAC-Address of the devices ring manager (MRM).
                  This variable contains the MAC-Address extracted from the
                  periodically received MRP_Test frame sent by the ring manager.
                  If the ring managers MAC-Address is not known, the variable shall
                  contain ff-ff-ff-ff-ff-ff.
                  "
 ::= { mrpInterconnectionEntry 14 }

mrpInterconnectionPeerMRMMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                  Device-MAC-Address of devices interconnection peer's, connected
                  to the devices interlink port, ring manager.
                  This variable contains the MRM-MAC-Address of the peers ring manager
                  contained in the media link check frame received at the interconnection port.
                  If the ring managers MAC-Address is not known, the variable shall

```

```

        contain ff-ff-ff-ff-ff-ff.
    ::= { mrpInterconnectionEntry 15 }

mrpInterconnectionPeerRingDomainUUID OBJECT-TYPE
    SYNTAX      IEC62439UuidType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "
        The domain UUID of the ring the interconnection peer's, connected
        to the devices interlink port, ring.
        This variable contains the MRM_DomainUUID of the peers ring
        contained in the media link check frame received at the interconnection port.
        "
    ::= { mrpInterconnectionEntry 16 }

-- *****
-- conformance statements
-- *****
mrpRedundancyGroups OBJECT IDENTIFIER ::= { mrpConformance 1 }

mrpDomainBasicGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainIndex,
        mrpDomainID,
        mrpDomainName,
        mrpDomainAdminRole,
        mrpDomainOperRole,
        mrpDomainRingPort1,
        mrpDomainRingPort1State,
        mrpDomainRingPort2,
        mrpDomainRingPort2State,
        mrpDomainState,
        mrpDomainBlocked
    }
    STATUS      current
    DESCRIPTION "
        Parameters which are necessary to monitor status of
        MRP domains.
        "
    ::= { mrpRedundancyGroups 1 }

mrpDomainManagerGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainManagerPriority
    }
    STATUS      current
    DESCRIPTION "
        Parameters which are additional to monitor status of
        MRP domain managers.
        "
    ::= { mrpRedundancyGroups 2 }

mrpDomainBlockingCntrlGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainMRMReactOnLinkChange
    }
    STATUS      current
    DESCRIPTION "
        Parameters which are optional to monitor and
        control blocking behavior of MRP domains.
        "
    ::= { mrpRedundancyGroups 3 }

mrpDomainDiagGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainError,
        mrpDomainRingOpenCount,
        mrpDomainLastRingOpenChange,
        mrpDomainRoundTripDelayMax,
        mrpDomainRoundTripDelayMin,
        mrpDomainResetRoundTripDelays
    }
    STATUS      current
    DESCRIPTION "
        Parameters which are optional to monitor
        and control the diagnostic status of MRP domains.
        "
    ::= { mrpRedundancyGroups 4 }

```

```

mrpDomainAdvancedGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainVlanId
    }
    STATUS current
    DESCRIPTION
        "
        Parameters which are optional to monitor
        the status of MRP domains.
        "
    ::= { mrpRedundancyGroups 5 }

mrpInterconnectionBasicGroup OBJECT-GROUP
    OBJECTS {
        mrpInterconnectionName,
        mrpInterconnectionPort,
        mrpInterconnectionPortState,
        mrpInterconnectionAdminRole,
        mrpInterconnectionOperRole,
        mrpInterconnectionMode,
        mrpInterconnectionState
    }
    STATUS current
    DESCRIPTION
        "
        Parameters which are necessary to monitor
        the status of Media redundancy
        Interconnection domains.
        "
    ::= { mrpRedundancyGroups 6 }

mrpInterconnectionDiagGroup OBJECT-GROUP
    OBJECTS {
        mrpInterconnectionError,
        mrpInterconnectionOpenCount,
        mrpInterconnectionLastOpenChange,
        mrpInterconnectionPeerMAC,
        mrpInterconnectionOwnMRMMAC,
        mrpInterconnectionPeerMRMMAC,
        mrpInterconnectionPeerRingDomainUUID
    }
    STATUS current
    DESCRIPTION
        "
        Parameters which are optional to monitor
        the diagnostic status of Media
        redundancy Interconnection domains.
        "
    ::= { mrpRedundancyGroups 7 }

mrpInterconnectionAdvancedGroup OBJECT-GROUP
    OBJECTS {
        mrpInterconnectionVlanId
    }
    STATUS current
    DESCRIPTION
        "
        Parameters which are optional to monitor the
        status of Media redundancy Interconnection domains.
        "
    ::= { mrpRedundancyGroups 8 }

-- *****
-- compliance specifications
-- *****
mrpRedundancyCompliances OBJECT IDENTIFIER ::= { mrpConformance 2 }

mrpDomainBasicCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "
        Basic implementation requirements for MRP support.
        The agent shall support the monitoring of redundancy
        domains.
        "
    MODULE
    MANDATORY-GROUPS {
        mrpDomainBasicGroup
    }
    GROUP mrpDomainAdvancedGroup
    DESCRIPTION
        "
        For SNMP agents it is optional to support

```

```

                                advanced monitoring of redundancy domains.
                                "
 ::= { mrpRedundancyCompliances 1 }

mrpDomainManagerCompliance  MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION  "
                Manager implementation requirements for MRP
                support. The agent shall support the monitoring
                of redundancy domains.
                "

    MODULE
    MANDATORY-GROUPS {
        mrpDomainBasicGroup,
        mrpDomainManagerGroup
    }

    GROUP      mrpDomainBlockingCntrlGroup
    DESCRIPTION "
                For SNMP agents it is optional to support blocking control.
                "

    GROUP      mrpDomainDiagGroup
    DESCRIPTION "
                For SNMP agents it is optional to support
                monitoring of the diagnostic status of MRP domains.
                "

 ::= { mrpRedundancyCompliances 2 }

mrpInterconnectionCompliance  MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION  "
                Implementation requirements for MRP Interconnection support.
                The agent shall support the monitoring of
                Media redundancy Interconnection domains.
                "

    MODULE
    MANDATORY-GROUPS {
        mrpInterconnectionBasicGroup
    }

    GROUP      mrpInterconnectionAdvancedGroup
    DESCRIPTION "
                For SNMP agents it is optional to support
                advanced monitoring of Media redundancy
                Interconnection domains.
                "

    GROUP      mrpInterconnectionDiagGroup
    DESCRIPTION "
                For SNMP agents it is optional to support
                monitoring of the diagnostic status of Media
                redundancy Interconnection domains.
                "

 ::= { mrpRedundancyCompliances 3 }

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

```

10.3 MRP MIB with a management and monitoring view

NOTE When copying the MIB from a PDF document, all headers and footers are removed and the characters are converted to the character format that SNMP requires. Although care has been taken not to use illegal characters, some can appear in the final document due to the editing process.

```

-- *****
--      Management MIB definitions
-- *****

IEC-62439-2-MIB      DEFINITIONS:= BEGIN

-- *****

```

```
-- Imports
-- *****

IMPORTS
    MODULE-IDENTITY,
    OBJECT-IDENTITY,
    OBJECT-TYPE,
    TimeTicks,
    Counter32,
    Unsigned32,
    Integer32                FROM SNMPv2-SMI
    OBJECT-GROUP,
    MODULE-COMPLIANCE        FROM SNMPv2-CONF
    TEXTUAL-CONVENTION,
    DisplayString, RowStatus FROM SNMPv2-TC;
-- *****
-- Root OID
-- *****
iec62439  MODULE-IDENTITY
    LAST-UPDATED "201405090000Z" -- May 09, 2014
    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.
        This definitions specify the monitoring and management
        variant of a SNMP entity. If the SNMP entity only contains
        monitoring functionality, then IEC 62439 monitoring MIB
        definitions shall be used instead of this MIB module.
        An entity may support either the monitoring facet only
        or the management facet including the monitoring facet
        as a subset.
    "

    REVISION      "201405090000Z" -- May 09, 2014
    DESCRIPTION "
        Manager (auto) support has been added in mrpDomainAdminRole.
        Interconnection functionality added.
        Units of mrpDomainRoundTripDelayMax and mrpDomainRoundTripDelayMin
        have been corrected to milliseconds.
        mrpDomainOperRole and mrpDomainAdminRole data type have been set
        to INTEGER.
        Missing constructs in import area have been added.
        Module identity block has been put on top.
    "

    REVISION      "200711120000Z" -- November 12, 2007
    DESCRIPTION "
        The undefined use of the value 'badValue' for objects
        of the optional 'mrpDomainDiagGroup' has been removed
        from DESCRIPTION clause at the objects concerned.
        The role dependency of the objects 'mrpDomainState' and
        'mrpDomainError' in the DESCRIPTION clause removed.
    "

    REVISION      "200708240000Z" -- August 24, 2007
    DESCRIPTION "
        Final Draft Proposal version of the Network Management
        interface for the Media Redundancy Protocol (MRP) with
        the monitoring functionality and the additional
        management functionality.
    "

    ::= { iso std(0) 62439 }

-- *****
-- Declaration of IEC62439UuidType
-- *****
```

```

IEC62439UuidType ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION "
        The IEC 61158-5-10 defines the structure of
        the UUID as a data type numeric
        identifier 1025.
    "
    SYNTAX          OCTET STRING (SIZE (16))

-- *****
-- Redundancy Protocols
-- *****
mrp          OBJECT IDENTIFIER ::= { iec62439 1 }
prp          OBJECT IDENTIFIER ::= { iec62439 2 }
crp          OBJECT IDENTIFIER ::= { iec62439 3 }
brp          OBJECT IDENTIFIER ::= { iec62439 4 }

-- *****
-- objects groups of redundancy object identifiers
-- *****
mrpObjects   OBJECT IDENTIFIER ::= { mrp 1 }
mrpNotifications OBJECT IDENTIFIER ::= { mrp 2 }
mrpConformance OBJECT IDENTIFIER ::= { mrp 3 }

-- *****
-- Objects of the MRP Network Management
-- *****
mrpDomainTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF MrpDomainEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION "
        Every entry in this table contains information about
        a Media Redundancy Protocol (MRP) domain.

        Row entries can be created or deleted via
        SNMP operations. Therefore the MAX-ACCESS values
        of some Object Identifier are different from the
        values of the same Object Identifier in the
        pure monitoring facet.
    "
    ::= { mrpObjects 1 }

mrpDomainEntry OBJECT-TYPE
    SYNTAX          MrpDomainEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION "An entry in the mrpDomainTable."
    INDEX           { mrpDomainIndex }
    ::= { mrpDomainTable 1 }

MrpDomainEntry ::= SEQUENCE {
    mrpDomainIndex          Unsigned32,
    mrpDomainID             IEC62439UuidType,
    mrpDomainName           DisplayString,
    mrpDomainAdminRole      INTEGER,
    mrpDomainOperRole       INTEGER,
    mrpDomainRingPort1      Integer32,
    mrpDomainRingPort1State INTEGER,
    mrpDomainRingPort2      Integer32,
    mrpDomainRingPort2State INTEGER,
    mrpDomainState          BITS,
    mrpDomainError          BITS,
    mrpDomainBlocked        INTEGER,
    mrpDomainVlanId         Unsigned32,
    mrpDomainManagerPriority INTEGER,
    mrpDomainRingOpenCount  Counter32,
    mrpDomainLastRingOpenChange TimeTicks,
    mrpDomainRoundTripDelayMax Unsigned32,
    mrpDomainRoundTripDelayMin Unsigned32,
    mrpDomainResetRoundTripDelays INTEGER,
    mrpDomainMRMReactOnLinkChange INTEGER,
    mrpDomainRowStatus      RowStatus
}

mrpDomainIndex OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      not-accessible

```

```

STATUS          current
DESCRIPTION      "The index of the entry"
::= { mrpDomainEntry 1 }

mrpDomainID      OBJECT-TYPE
SYNTAX          IEC62439UuidType
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION      "
                Universally unique identifier belongs to the MRP domain
                which represents a ring.
                "
::= { mrpDomainEntry 2 }

mrpDomainName    OBJECT-TYPE
SYNTAX          DisplayString
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION      "
                A logical name for the MRP domain to ease the
                management of MRP domains.
                "
::= { mrpDomainEntry 3 }

mrpDomainAdminRole OBJECT-TYPE
SYNTAX          INTEGER {
                disable(0),
                client(1),
                manager(2),
                managerAutoComp(3),
                managerAuto(4)
                }
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION      "
                Control the MRP behavior of the system per domain.

                If the value is set to disable(0) the MRP entity of
                this domain shall be disabled.
                If the value is set to client(1) the entity shall be set
                to the role of a Media Redundancy Client (MRC).
                If the value is set to manager(2) the entity shall be
                set to the role of a Media Redundancy Manager (MRM).
                If the value is set to managerAutoComp(3) the entity shall be
                set to the role of a Media Redundancy Manager Auto (MRA)
                complying to Annex A.
                If the value is set to managerAuto(4) the entity shall be
                set to the role of a Media Redundancy Manager Auto (MRA) not
                supporting Annex A.
                The factory settings are recommended to adjust the
                value of this object to the client(1) capability of the
                component, or, if supported, to the managerAutoComp(3)
                or managerAuto(4) capability,
                in order to prevent multiple managers are
                in ring (the order of the capabilities are not
                necessarily conform to the order of the object values
                here).
                If the agent restricts the write access, no matter what
                reason, it shall reject write requests by responding
                with 'badValue'.
                "
::= { mrpDomainEntry 4 }

mrpDomainOperRole OBJECT-TYPE
SYNTAX          INTEGER {
                disable(0),
                client(1),
                manager(2),
                managerAutoComp(3),
                managerAuto(4)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION      "
                The operational role of an MRP entity per domain.
                A value of disabled(0) signals that the entity doesn't
                work (whatever reason).
                A value of client(1) signals that the entity is in a

```

```

        client role.
        A value of manager(2) signals that the entity is the
        manager of this MRP domain.
        A value of managerAutoComp(3) signals that the entity is
        in automanager role complying to Annex A.
        A value of managerAuto(4) signals that the entity is in
        automanager role not supporting Annex A.
        "
 ::= { mrpDomainEntry 5 }

mrpDomainRingPort1 OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
        The ifIndex of the layer 2 interface which is used
        as ring port 1.
        "
 ::= { mrpDomainEntry 6 }

mrpDomainRingPort1State OBJECT-TYPE
    SYNTAX      INTEGER {
        disabled(1),
        blocked(2),
        forwarding(3),
        not-connected(4)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Operational state of the first Ring-Port.

        disabled(1)      indicates that all frames are dropped.
        blocked(2)       indicates that all frames are dropped except
                        - MRP topology change frames and MRP test frames from
                          a MRM,
                        - MRP link change frames from an MRC,
                        - MRP interconnection topology change from a MIM,
                        - MRP interconnection link change from a MIC,
                        - frames from other protocols that also define to pass
                          blocked(2) ports.
        forwarding(3)    indicates that all frames are passed through
                        according to the forwarding behavior
                        of IEEE 802.1D.
        not-connected(4) indicates that the port has no link.
        "
 ::= { mrpDomainEntry 7 }

mrpDomainRingPort2 OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
        The ifIndex of the layer 2 interface which is
        used as ring port 2.
        "
 ::= { mrpDomainEntry 8 }

mrpDomainRingPort2State OBJECT-TYPE
    SYNTAX      INTEGER {
        disabled(1),
        blocked(2),
        forwarding(3),
        not-connected(4)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Operational state of the second Ring-Port.

        disabled(1)      indicates that all frames are dropped.
        blocked(2)       indicates that all frames are dropped except
                        - MRP topology change frames and MRP test frames from
                          a MRM,
                        - MRP link change frames from an MRC,
                        - MRP interconnection topology change from a MIM,
                        - MRP interconnection link change from a MIC,

```

```

        - frames from other protocols that also define to pass
          blocked(2) ports.
forwarding(3)    indicates that all frames are passed through
                  according to the forwarding behavior
                  of IEEE 802.1D.
not-connected(4) indicates that the port has no link.
"
 ::= { mrpDomainEntry 9 }

mrpDomainState OBJECT-TYPE
SYNTAX          BITS {
                  disabled(0),
                  undefined(1),
                  ringOpen(2),
                  reserved(3)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "
                  Operational status of the MRP entity.
disabled(0) - MRP switched off. All higher bits are invalid
              and shall be reset.
undefined(1) - Value is not valid.
               All higher bits are invalid and shall be reset.
ringOpen(2) - MRP ring redundancy lost
               All higher bits are invalid and shall be reset.
reserved(3) - reserved for further extensions.
"
 ::= { mrpDomainEntry 10 }

mrpDomainError OBJECT-TYPE
SYNTAX          BITS {
                  noError(0),
                  invalidVlanId(1),
                  invalid(2),
                  multipleMRM(3),
                  singleSideReceive(4)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "
                  If the device couldn't switch to the desired administrative state
                  (thus the value of 'mrpDomainState' is not the expected one),
                  this value provides the reason. Otherwise the bit noError(0) is set.

noError(0)      - the operational state of the device is conform
                  to administrative state.
                  All higher bits are invalid and shall be reset.
invalidVlanId(1) - the assigned VLAN ID is not permitted .
invalid (2)     - Value is not valid.
                  All higher bits are invalid and shall be reset.
multipleMRM(3)  - multiple active managers in ring domain.
singleSideReceive(4) - the test frames of an MRM
                  have been seen, but only on one port.
"
 ::= { mrpDomainEntry 11 }

mrpDomainBlocked OBJECT-TYPE
SYNTAX          INTEGER {
                  enabled(1),
                  disabled(2)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "
                  The entity supports blocked ring ports.
                  Shows whether a mrp domain requires the support of the
                  BLOCKED port state at ring ports.
                  The manager of a mrp domain decides whether this support
                  is required.
                  Set to enabled the manager demands that all clients
                  shall support the blocked attribute also set to enabled.
                  If mrpDomainBlocked is set disabled at the manager,
                  then the value of mrpDomainBlocked can be arbitrary
                  at the clients.

enabled(1)      client: supports ring ports whose port state can be

```

```

        blocked.
        manager: works only with clients supporting blocked
        ring ports.
disabled(2) client: no support of blocked ring ports.
        manager: Work with clients supporting blocked ring
        ports and with clients not supporting
        blocked ring ports.
"
 ::= { mrpDomainEntry 12 }

mrpDomainVlanId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
        The VLAN ID assigned to the MRP protocol.
        The VLAN ID only is in use when the bit invalidVlanId
        is not set in mrpDomainError.
        If value is set to 0 no VLAN is assigned.
        The invalidVlanId shall be set to 0 (no Error).
        "
 ::= { mrpDomainEntry 13 }

-- *****
-- Objects of the MRP Network Management for manager role
-- *****
mrpDomainManagerPriority OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
        The priority of this MRP entity. If the device is
        client only, the value of this object shall be ignored
        by the MRP entity. Only the four most significant bits
        shall be used, the bits 0 to 11 are reserved. The
        smaller value has the higher priority.
        "
 ::= { mrpDomainEntry 14 }

mrpDomainRingOpenCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Counter for ring-state changes to open.
        "
 ::= { mrpDomainEntry 15 }

mrpDomainLastRingOpenChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Timeticks since last change of ring-state
        to ring open.
        "
 ::= { mrpDomainEntry 16 }

mrpDomainRoundTripDelayMax OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The max. Round-Trip-Delay (in milliseconds)
        which was measured since startup.
        "
 ::= { mrpDomainEntry 17 }

mrpDomainRoundTripDelayMin OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The min. Round-Trip-Delay (in milliseconds)
        which was measured since startup.
        "

```

```

 ::= { mrpDomainEntry 18 }

mrpDomainResetRoundTripDelays OBJECT-TYPE
    SYNTAX      INTEGER {
        resetDelays(1)
    }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION   "
        A write request with resetDelays(1) shall reset
        the values of mrpDomainRoundTripDelayMax and
        mrpDomainRoundTripDelayMin to zero at the
        same time.
    "

 ::= { mrpDomainEntry 19 }

mrpDomainMRMReactOnLinkChange OBJECT-TYPE
    SYNTAX      INTEGER {
        enabled(1),
        disabled(2)
    }
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION   "
        Tells whether the MRM reacts on link down MRP link change
        frames.
        enabled(1) indicates that the MRM reacts immediately on link down MRP
        link change frames.
        disabled(2) indicates that the MRM does not react on
        link down MRP link change frames.
    "

 ::= { mrpDomainEntry 20 }

-- *****
-- additional Objects of the MRP Network Management for management
-- *****

mrpDomainRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION   "
        Support of creation and deletion of rows.
        A row can be taken out of service by writing
        the value notInService(2).
    "

 ::= { mrpDomainEntry 21}

-- *****
-- Objects of the Media redundancy Interconnection
-- *****

mrpInterconnectionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MrpInterconnectionEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION   "
        Every entry in this table contains information about
        a Media redundancy Interconnection domain.

        Row entries can be created or deleted via
        SNMP operations. Therefore the MAX-ACCESS values
        of some Object Identifier are different from the
        values of the same Object Identifier in the
        pure monitoring facet.
    "

 ::= { mrpObjects 2 }

mrpInterconnectionEntry OBJECT-TYPE
    SYNTAX      MrpInterconnectionEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION   "An entry in the mrpInterconnectionTable."
    INDEX       { mrpInterconnectionID }
 ::= { mrpInterconnectionTable 1 }

MrpInterconnectionEntry ::= SEQUENCE {
    mrpInterconnectionID      Unsigned32,
    mrpInterconnectionName    DisplayString,

```

```

        mrpInterconnectionPort          Integer32,
        mrpInterconnectionPortState     INTEGER,
        mrpInterconnectionVlanId        Unsigned32,
        mrpInterconnectionMode          INTEGER,
        mrpInterconnectionAdminRole     INTEGER,
        mrpInterconnectionOperRole     INTEGER,
        mrpInterconnectionState         INTEGER,
        mrpInterconnectionError         BITS,
        mrpInterconnectionOpenCount     Counter32,
        mrpInterconnectionLastOpenChange TimeTicks,
        mrpInterconnectionPeerMAC       OCTET STRING,
        mrpInterconnectionOwnMRMMAC     OCTET STRING,
        mrpInterconnectionPeerMRMMAC    OCTET STRING,
        mrpInterconnectionPeerRingDomainUUID IEC62439UuidType,
        mrpInterconnectionRowStatus     RowStatus
    }

mrpInterconnectionID OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION  "The index of the entry"
    ::= { mrpInterconnectionEntry 1 }

mrpInterconnectionName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION  "
        A logical name for the Media redundancy Interconnection domain to
        ease the management of Media redundancy Interconnection domains.
    "
    ::= { mrpInterconnectionEntry 2 }

mrpInterconnectionPort OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION  "
        The ifIndex of the layer 2 interface which is used
        as interconnection port.
    "
    ::= { mrpInterconnectionEntry 3 }

mrpInterconnectionPortState OBJECT-TYPE
    SYNTAX      INTEGER {
        disabled(1),
        blocked(2),
        forwarding(3),
        notConnected(4)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION  "
        Operational state of the Interconnection port.

        disabled(1) indicates that all frames are dropped.
        blocked(2) indicates that all frames are dropped except
            -MRP Interconnection topology change frames,
            -MRP Interconnection test frames,
            -MRP Interconnection link change frames,
            -MRP Interconnection link status poll frames,
            -frames from other protocols that also define to pass
              blocked(2) ports.
        forwarding(3) indicates that all frames are passed through
            according to the forwarding behavior of IEEE 802.1D.
        notConnected(4) indicates that the port has no link.
    "
    ::= { mrpInterconnectionEntry 4 }

mrpInterconnectionVlanId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION  "
        The VLAN ID assigned to the MRP Interconnection protocol.
        The VLAN ID only is in use when mrpInterconnectionError
        does not signal invalidVlanId.
    "

```

```

        If value is set to 0 no VLAN is assigned.
        "
 ::= { mrpInterconnectionEntry 5 }

mrpInterconnectionMode OBJECT-TYPE
    SYNTAX      INTEGER {
                    undefined(1),
                    lcMode(2),
                    rcMode(3)
                }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "
        Control the MRP Interconnection mode.

        The value undefined(1) signals that the interconnection mode is
        neither set to lcMode(2) nor to rcMode(3).
        If the value is set to lcMode(2) the device in Media redundancy
        Interconnection Manager (MIM) role is using the link check mode to
        detect open or closed interconnection links.
        If the value is set to rcMode(3) the device in Media redundancy
        Interconnection Manager (MIM) role is using the ring check mode to
        detect open or closed interconnection links.

        The factory settings are recommended to adjust the
        value of this object to the lcMode(2) capability.
        If the agent restricts the write access, no matter what
        reason, it shall reject write requests by responding
        with 'badValue'.
        "

 ::= { mrpInterconnectionEntry 6 }

mrpInterconnectionAdminRole OBJECT-TYPE
    SYNTAX      INTEGER {
                    disable(0),
                    client(1),
                    manager(2)
                }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "
        Control the MRP Interconnection behavior of the system per
        Media redundancy Interconnection domain.

        If the value is set to disable(0) the MRP Interconnection
        entity of this domain shall be disabled.
        If the value is set to client(1) the entity shall be set
        to the role of a Media redundancy Interconnection Client (MIC).
        If the value is set to manager(2) the entity shall be set
        to the role of a Media redundancy Interconnection Manager (MIM).
        The factory settings are recommended to adjust the
        value of this object to the client(1) capability of the
        component in order to prevent multiple managers are
        in the interconnection (the order of the capabilities are not
        necessarily conform to the order of the object values
        here).
        If the agent restricts the write access, no matter what
        reason, it shall reject write requests by responding
        with 'badValue'.
        "

 ::= { mrpInterconnectionEntry 7 }

mrpInterconnectionOperRole OBJECT-TYPE
    SYNTAX      INTEGER {
                    disable(0),
                    client(1),
                    manager(2)
                }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The operational role of an MRP Interconnection entity
        per Media redundancy Interconnection domain.
        A value of disabled(0) signals that the entity doesn't
        work (whatever reason).
        A value of client(1) signals that the entity is in a
        Media redundancy Interconnection Client role.
        A value of manager(2) signals that the entity is the
    
```

```

        manager of this Media redundancy Interconnection domain.
    "
 ::= { mrpInterconnectionEntry 8 }

mrpInterconnectionState OBJECT-TYPE
    SYNTAX      INTEGER {
        disable(0),
        undefined(1),
        interconnOpen(2),
        interconnClosed(3)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Operational status of the MRP Interconnection entity.
        disable(0)      - The entity is disabled.
        undefined(1)    - Value is not valid.
        interconnOpen(2) - MRP Interconnection redundancy lost of this Media
                        redundancy Interconnection domain.
        interconnClosed(3) - MRP Interconnection redundancy available of this Media
                        redundancy Interconnection domain.
    "
 ::= { mrpInterconnectionEntry 9 }

mrpInterconnectionError OBJECT-TYPE
    SYNTAX      BITS {
        noError(0),
        invalidVlanId(1)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Configuration status of the MRP Interconnection entity.

        noError(0)      - no configuration error detected.
        invalidVlanId(1) - the assigned VLAN ID is not permitted.
    "
 ::= { mrpInterconnectionEntry 10 }

mrpInterconnectionOpenCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Counter for MRP Interconnection state changes to
        interconnOpen.
    "
 ::= { mrpInterconnectionEntry 11 }

mrpInterconnectionLastOpenChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Timeticks since last change of
        mrpInterconnectionState to interconnOpen.
    "
 ::= { mrpInterconnectionEntry 12 }

mrpInterconnectionPeerMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Device-MAC-Address of the devices interconnection peer, connected
        to the devices interlink port. This variable contains
        the MAC-Address contained in the media link check frame received at the
        interconnection port.
    "
 ::= { mrpInterconnectionEntry 13 }

mrpInterconnectionOwnMRMMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The device-MAC-Address of the devices ring manager (MRM).
        This variable contains the MAC-Address extracted from the
        periodically received MRP_Test frame sent by the ring manager.
        If the ring managers MAC-Address is not known, the variable shall
        contain ff-ff-ff-ff-ff-ff.
    "

```

```

"
::= { mrpInterconnectionEntry 14 }

mrpInterconnectionPeerMRMMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
                Device-MAC-Address of devices interconnection peer's, connected
                to the devices interlink port, ring manager.
                This variable contains the MRM-MAC-Address of the peers ring manager
                contained in the media link check frame received at the interconnection port.
                If the ring managers MAC-Address is not known, the variable shall
                contain ff-ff-ff-ff-ff-ff.
                "

::= { mrpInterconnectionEntry 15 }

mrpInterconnectionPeerRingDomainUUID OBJECT-TYPE
    SYNTAX      IEC62439UuidType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
                The domain UUID of the ring the interconnection peer's, connected
                to the devices interlink port, ring.
                This variable contains the MRM_DomainUUID of the peers ring
                contained in the media link check frame received at the interconnection port.
                "

::= { mrpInterconnectionEntry 16 }

mrpInterconnectionRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
                Support of creation and deletion of rows.
                A row can be taken out of service by writing
                the value notInService(2).
                "

::= { mrpInterconnectionEntry 18 }

-- *****
-- conformance statements
-- *****
mrpRedundancyGroups OBJECT IDENTIFIER ::= { mrpConformance 1 }

mrpDomainBasicGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainIndex,
        mrpDomainID,
        mrpDomainName,
        mrpDomainAdminRole,
        mrpDomainOperRole,
        mrpDomainRingPort1,
        mrpDomainRingPort1State,
        mrpDomainRingPort2,
        mrpDomainRingPort2State,
        mrpDomainState,
        mrpDomainBlocked,
        mrpDomainRowStatus
    }
    STATUS      current
    DESCRIPTION  "
                Parameters which are necessary to monitor
                and control the status of MRP domains.
                "

::= { mrpRedundancyGroups 1 }

mrpDomainManagerGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainManagerPriority
    }
    STATUS      current
    DESCRIPTION  "
                Parameters which are additional to monitor
                and control the status of MRP domain managers.
                "

::= { mrpRedundancyGroups 2 }

mrpDomainBlockingCntlGroup OBJECT-GROUP
    OBJECTS {

```

```

        mrpDomainMRMReactOnLinkChange
    }
    STATUS          current
    DESCRIPTION      "
                    Parameters which are optional to monitor and control
                    blocking behavior of MRP domains.
                    "

 ::= { mrpRedundancyGroups 3 }

mrpDomainDiagGroup OBJECT-GROUP
OBJECTS {
    mrpDomainError,
    mrpDomainRingOpenCount,
    mrpDomainLastRingOpenChange,
    mrpDomainRoundTripDelayMax,
    mrpDomainRoundTripDelayMin,
    mrpDomainResetRoundTripDelays
}
STATUS          current
DESCRIPTION      "
                    Parameters which are optional to monitor
                    and control the diagnostic status of MRP domains.
                    "

 ::= { mrpRedundancyGroups 4 }

mrpDomainAdvancedGroup OBJECT-GROUP
OBJECTS {
    mrpDomainVlanId
}
STATUS          current
DESCRIPTION      "
                    Parameters which are optional to monitor and control
                    the status of MRP domains.
                    "

 ::= { mrpRedundancyGroups 5 }

mrpInterconnectionBasicGroup OBJECT-GROUP
OBJECTS {
    mrpInterconnectionName,
    mrpInterconnectionPort,
    mrpInterconnectionPortState,
    mrpInterconnectionAdminRole,
    mrpInterconnectionOperRole,
    mrpInterconnectionMode,
    mrpInterconnectionState,
    mrpInterconnectionRowStatus
}
STATUS          current
DESCRIPTION      "
                    Parameters which are necessary to monitor
                    and control the status of Media redundancy
                    Interconnection domains.
                    "

 ::= { mrpRedundancyGroups 6 }

mrpInterconnectionDiagGroup OBJECT-GROUP
OBJECTS {
    mrpInterconnectionError,
    mrpInterconnectionOpenCount,
    mrpInterconnectionLastOpenChange,
    mrpInterconnectionPeerMAC,
    mrpInterconnectionOwnMRMMAC,
    mrpInterconnectionPeerMRMMAC,
    mrpInterconnectionPeerRingDomainUUID
}
STATUS          current
DESCRIPTION      "
                    Parameters which are optional to monitor
                    and control the diagnostic status of Media
                    redundancy Interconnection domains.
                    "

 ::= { mrpRedundancyGroups 7 }

mrpInterconnectionAdvancedGroup OBJECT-GROUP
OBJECTS {
    mrpInterconnectionVlanId
}
STATUS          current

```

```

DESCRIPTION      "
                  Parameters which are optional to monitor and control
                  the status of Media redundancy Interconnection domains.
                  "
 ::= { mrpRedundancyGroups 8 }

-- *****
-- compliance specifications
-- *****
mrpRedundancyCompliances OBJECT IDENTIFIER ::= { mrpConformance 2 }

mrpDomainBasicCompliance      MODULE-COMPLIANCE
STATUS                        current
DESCRIPTION                  "
                            Basic implementation requirements for MRP support.
                            The agent shall support the monitoring and control
                            of redundancy domains.
                            "

MODULE
MANDATORY-GROUPS {
    mrpDomainBasicGroup
}

GROUP    mrpDomainAdvancedGroup
DESCRIPTION "
        For SNMP agents it is optional to support
        advanced monitoring and control of
        redundancy domains.
        "
 ::= { mrpRedundancyCompliances 1 }

mrpDomainManagerCompliance      MODULE-COMPLIANCE
STATUS                        current
DESCRIPTION                  "
                            Manager implementation requirements for MRP
                            support. The agent shall support the monitoring
                            and control of redundancy domains.
                            "

MODULE
MANDATORY-GROUPS {
    mrpDomainBasicGroup,
    mrpDomainManagerGroup
}

GROUP    mrpDomainBlockingCntrlGroup
DESCRIPTION "
        For SNMP agents it is optional to support blocking control.
        "

GROUP    mrpDomainDiagGroup
DESCRIPTION "
        For SNMP agents it is optional to support
        monitoring of the diagnostic status of MRP domains.
        "
 ::= { mrpRedundancyCompliances 2 }

mrpInterconnectionCompliance      MODULE-COMPLIANCE
STATUS                        current
DESCRIPTION                  "
                            Implementation requirements for MRP Interconnection support.
                            The agent shall support the monitoring and control
                            of Media redundancy Interconnection domains.
                            "

MODULE
MANDATORY-GROUPS {
    mrpInterconnectionBasicGroup
}

GROUP    mrpInterconnectionAdvancedGroup
DESCRIPTION "
        For SNMP agents it is optional to support
        advanced monitoring and control of Media
        redundancy Interconnection domains.
        "

GROUP    mrpInterconnectionDiagGroup
DESCRIPTION "

```

For SNMP agents it is optional to support
monitoring of the diagnostic status of Media
redundancy Interconnection domains.
"

::= { mrpRedundancyCompliances 3 }

END

-- *****
-- EOF
-- *****

Annex A (normative)

Optional earlier version of the Automanager protocol

A device may support backwards compatibility to the earlier version of the Automanager protocol. In order to do so, it shall implement the MRA functionality as defined in the Clauses 5, 6, 7 and 8 and in addition:

- Receive, analyse and transmit MRA MRP_Test frames containing the MRP_Option data content given in Table A.1:

Table A.1 – Compatible mode MRP_Option for MRP_Test Substitutions

Substitution name	Structure
MRP_Option	MRP_TLVHeader (Type=0x7F), MRP_OUI (=0x08-00-06), MRP_SubOption1, [MRP_SubOption2]
MRP_SubOption1	MRP_Ed1Type (=0x00), MRP_Ed1ManufacturerData (=0x00, 0x00)
MRP_SubOption2	MRP_SubTLVHeader (Type=0x03, Length = 0)

- Receive, analyse and transmit MRA MRP_TestMgrNAck and MRP_TestPropagate frames containing the MRP_Option data content given in Table A.2:

Table A.2 – Compatible mode MRP_Option frames MRP_TestMgrNAck and MRP_TestPropagate Substitutions

Substitution name	Structure
MRP_Option	MRP_TLVHeader (Type=0x7F, Length = 0x1A), MRP_OUI (=0x08-00-06), MRP_SubOption1, MRP_SubOption2
MRP_SubOption1	MRP_Ed1Type (=0x00), MRP_Ed1ManufacturerData (=0x00, 0x00)
MRP_SubOption2	MRP_SubTLVHeader (Type=0x01 or 0x02, Length = 0x12), MRP_Prio, MRP_SA, MRP_OtherMRMPrio, MRP_OtherMRMSA, Padding (=0x00, 0x00)

Bibliography

The referenced documents are listed in IEC 62439-1. In addition the following documents are referenced.

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

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

SOMMAIRE

AVANT-PROPOS.....	169
INTRODUCTION.....	171
1 Domaine d'application.....	173
2 Références normatives	173
3 Termes, définitions, abréviations, acronymes et conventions	174
3.1 Termes et définitions.....	174
3.2 Abréviations et acronymes	174
3.3 Conventions.....	175
4 Vue d'ensemble du MRP.....	175
5 Comportement de redondance du support de MRP	179
5.1 Généralités	179
5.2 Ports en anneau.....	179
5.3 Gestionnaire de redondance du support (MRM)	180
5.4 Client de redondance du support (MRC)	182
5.5 Domaine de redondance.....	182
5.6 Vérification de liaison de support	182
5.7 Application du protocole de vérification de la continuité.....	183
5.7.1 Généralités	183
5.7.2 Intervalle de message de vérification de la continuité.....	183
5.7.3 Niveau du domaine de maintenance	183
5.7.4 ID d'association de maintenance (MAID, Maintenance Association ID)	183
5.7.5 ID de point d'extrémité d'association de maintenance (MEPID)	184
5.7.6 TLV d'ID d'émetteur	184
5.7.7 TLV d'état de port	184
5.7.8 TLV d'état d'interface	184
5.8 Utilisation avec des diagnostics et des alarmes.....	184
5.9 Diagnostic de l'anneau	184
5.10 Plusieurs MRM dans un seul anneau: Option de vote du gestionnaire	185
5.10.1 Généralités	185
5.10.2 Principe de base du processus de sélection du gestionnaire	187
5.10.3 Le processus de sélection du gestionnaire.....	187
5.11 BLOCKED non pris en charge (option).....	190
5.12 Port d'interconnexion.....	190
5.13 Gestionnaire d'interconnexion de redondance du support (MIM)	191
5.14 Client d'interconnexion de redondance du support (MIC).....	194
5.15 Domaine d'interconnexion	195
5.16 Diagnostic d'interconnexion.....	195
6 Spécification de classe du MRP	195
6.1 Généralités	195
6.2 Modèle.....	196
6.2.1 Modèle de redondance du support.....	196
6.2.2 Modèle d'interconnexion de redondance du support.....	197
6.3 Attributs	197
7 Spécification de service du MRP	201
7.1 Démarrage MRM.....	201

7.2	Arrêt MRM	203
7.3	Modification de l'état	204
7.4	Démarrage MRC	204
7.5	Arrêt MRC.....	206
7.6	Lecture MRM	206
7.7	Lecture MRC.....	208
7.8	Démarrage MIM	210
7.9	Arrêt MIM.....	211
7.10	Modification de l'état d'interconnexion	212
7.11	Démarrage MIC.....	213
7.12	Arrêt MIC	214
7.13	Lecture MIM.....	215
7.14	Lecture MIC	217
8	Spécification du protocole MRP	219
8.1	Description du PDU.....	219
8.1.1	Types de données de base.....	219
8.1.2	Référence de syntaxe abstraite du DLPDU	219
8.1.3	Codage du champ de DLPDU SourceAddress.....	219
8.1.4	Codage du champ de DLPDU DestinationAddress	220
8.1.5	Codage du champ TagControlInformation	220
8.1.6	Codage du champ LT.....	221
8.1.7	Syntaxe abstraite des APDU du MRP	221
8.1.8	Codage du champ MRP_TLVHeader	222
8.1.9	Codage du champ MRP_SubTLVHeader	223
8.1.10	Codage des champs MRP_Ed1Type et MRP_Ed1ManufacturerData.....	223
8.1.11	Codage du champ MRP_Version	224
8.1.12	Codage du champ MRP_SequenceID	224
8.1.13	Codage du champ MRP_SA	224
8.1.14	Codage du champ MRP_OtherMRMSA.....	224
8.1.15	Codage du champ MRP_Prio.....	225
8.1.16	Codage du champ MRP_OtherMRMPrio	225
8.1.17	Codage du champ MRP_PortRole	225
8.1.18	Codage du champ MRP_RingState.....	226
8.1.19	Codage du champ MRP_Interval	226
8.1.20	Codage du champ MRP_Transition.....	226
8.1.21	Codage du champ MRP_TimeStamp	226
8.1.22	Codage du champ MRP_Blocked.....	227
8.1.23	Codage du champ MRP_ManufacturerOUI.....	227
8.1.24	Codage du champ MRP_IECOUI	227
8.1.25	Codage du champ MRP_ManufacturerData	227
8.1.26	Codage du champ MRP_DomainUUID	227
8.1.27	Codage du champ MRP_InState.....	227
8.1.28	Codage du champ MRP_InID	228
8.2	Machines de protocole	228
8.2.1	Machine de protocole du MRM	228
8.2.2	Machine de protocole du MRC.....	239
8.2.3	Machine de protocole du MRA.....	245
8.2.4	Fonctions des MRA, MRM et MRC.....	263
8.2.5	Temporisateur d'annulation de la FDB	267

8.2.6	Temporisateur de changement de topologie.....	267
8.2.7	Machine de protocole du MIM.....	268
8.2.8	Machine de protocole du MIC	277
8.2.9	Fonctions du MIM et du MIC.....	285
8.2.10	Temporisateur de changement de topologie d'interconnexion.....	289
8.2.11	Temporisateur d'interrogation sur l'état de liaison d'interconnexion	289
9	Installation, configuration et réparation de MRP	290
9.1	Paramètres du port d'interconnexion et du port en anneau	290
9.2	Paramètres de topologie en anneau	290
9.3	Paramètres du MRM	290
9.4	Paramètres et contraintes pour le MRC	291
9.5	Compatibilité du MRA avec des versions antérieures du protocole du gestionnaire automatique	292
9.6	Paramètres de topologie d'interconnexion.....	292
9.7	Paramètres du MIM.....	292
9.8	Paramètres et contraintes pour le MIC.....	293
9.9	Calcul du temps de reprise de l'anneau du MRP	293
9.9.1	Vue d'ensemble	293
9.9.2	Déduction de formule	293
9.9.3	Calcul du scénario le plus défavorable pour un temps de reprise de 10 ms	295
9.9.4	Calcul du scénario le plus défavorable pour 50 appareils	296
9.10	Calcul du temps de sélection du gestionnaire automatique du MRP.....	297
10	Base d'informations de gestion du MRP (MIB)	297
10.1	Généralités	297
10.2	MIB du MRP avec vue de surveillance	297
10.3	MIB du MRP avec vue de gestion et de surveillance	310
Annexe A (normative) Versions antérieures facultatives du protocole du gestionnaire automatique		324
Bibliographie		325
Figure 1 – Deux anneaux de MRP connectés de manière redondante via l'interconnexion de MRP		177
Figure 2 – MRP en pile		179
Figure 3 – Topologie en anneau du MRP avec un gestionnaire et des clients		180
Figure 4 – Anneau ouvert du MRP avec le MRM.....		181
Figure 5 – Anneau de MRP avec un MRA au démarrage du réseau.....		186
Figure 6 – Anneau de MRP après le processus de sélection du gestionnaire.....		186
Figure 7 – Processus de sélection du gestionnaire		189
Figure 8 – MRA situé hors de l'anneau de MRP.....		190
Figure 9 – Topologie d'interconnexion du MRP		192
Figure 10 – Interconnexion en anneau du MRP ouverte		193
Figure 11 – Machine de protocole MRP pour le MRM.....		228
Figure 12 – Machine de protocole MRP pour le MRC.....		239
Figure 13 – Machine de protocole MRP pour le MRA		246
Figure 14 – Machine de protocole MRP pour le MIM en RC-mode et LC-mode		269
Figure 15 – Machine de protocole MRP pour le MIC en RC-mode et LC-mode.....		278

Tableau 1 – Informations de brevet	171
Tableau 2 – Codage de l'ID d'association de maintenance (MAID)	184
Tableau 3 – Service Start MRM du MRP	202
Tableau 4 – Service Stop MRM du MRP	203
Tableau 5 – Service Change State du MRP	204
Tableau 6 – Service Start MRC du MRP	205
Tableau 7 – Service Arrêt MRC du MRP	206
Tableau 8 – Service Read MRM du MRP	207
Tableau 9 – Service Read MRC du MRP	209
Tableau 10 – Service Start MIM du MRP	210
Tableau 11 – Service Stop MIM du MRP	212
Tableau 12 – Service Interconnection Change State du MRP	212
Tableau 13 – Service Start MIC du MRP	213
Tableau 14 – Service Stop MIC du MRP	214
Tableau 15 – Service Read MIM du MRP	216
Tableau 16 – Service Read MIC du MRP	218
Tableau 17 – Syntaxe du DLPDU du MRP pour l'ISO/IEC/IEEE 8802-3 (IEEE 802.3)	219
Tableau 18 – OUI du MRP	220
Tableau 19 – MulticastMACAddress du MRP	220
Tableau 20 – Champ TagControlInformation.Priority du MRP	221
Tableau 21 – Champ LT du MRP	221
Tableau 22 – Syntaxe des APDU du MRP	221
Tableau 23 – Substitutions au sein du MRP	222
Tableau 24 – MRP_TLVHeader.Type	223
Tableau 25 – MRP_SubTLVHeader.Type	223
Tableau 26 – MRP_Ed1Type et MRP_Ed1ManufacturerData	224
Tableau 27 – MRP_Ed1Type et MRP_Ed1ManufacturerData	224
Tableau 28 – MRP_Version	224
Tableau 29 – Codage du champ MRP_OtherMRMSA	225
Tableau 30 – MRP_Prio	225
Tableau 31 – Codage du champ MRP_OtherMRMPrio	225
Tableau 32 – MRP_PortRole	225
Tableau 33 – MRP_RingState	226
Tableau 34 – MRP_Interval	226
Tableau 35 – MRP_Transition	226
Tableau 36 – MRP_TimeStamp	226
Tableau 37 – MRP_Blocked	227
Tableau 38 – MRP_DomainUUID	227
Tableau 39 – MRP_InState	228
Tableau 40 – Variables locales du MRP de la machine de protocole du MRM	230
Tableau 41 – Diagramme d'états du MRM	231
Tableau 42 – Variables locales du MRP de la machine de protocole du MRC	240

Tableau 43 – Diagramme d'états du MRC.....	241
Tableau 44 – Variables locales du MRP de la machine de protocole du MRA	247
Tableau 45 – Diagramme d'états du MRA.....	248
Tableau 46 – Fonctions et macros du MRP.....	264
Tableau 47 – Temporisateur d'annulation de la FDB de filtrage du MRP.....	267
Tableau 48 – Temporisateur de changement de topologie du MRP	268
Tableau 49 – Variables locales du MRP de la machine de protocole du MIM	270
Tableau 50 – Diagramme d'états du MIM en LC-mode	271
Tableau 51 – Diagramme d'états du MIM en RC-mode.....	274
Tableau 52 – Variables locales du MRP de la machine de protocole du MIC	279
Tableau 53 – Diagramme d'états du MIC en LC-mode	280
Tableau 54 – Diagramme d'états du MIC en RC-mode	283
Tableau 55 – Fonctions d'interconnexion du MRP.....	286
Tableau 56 – Temporisateur de changement de topologie d'interconnexion du MRP.....	289
Tableau 57 – Temporisateur d'interrogation sur l'état de liaison d'interconnexion du MRP	290
Tableau 58 – Paramètres de réseau/connexion du MRP	290
Tableau 59 – Paramètres relatifs au MRM du MRP	291
Tableau 60 – Paramètres relatifs au MRC du MRP	291
Tableau 61 – Paramètres relatifs au MIM du MRP	293
Tableau 62 – Paramètres relatifs au MIC du MRP.....	293
Tableau A.1 – MRP_Option en mode compatible pour les remplacements MRP_Test.....	324
Tableau A.2 – Cadres MRP_Option en mode compatible pour les remplacements MRP_TestMgrNAck et MRP_TestPropagate	324

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

RÉSEAUX DE COMMUNICATION INDUSTRIELS –
RÉSEAUX D'AUTOMATISME A HAUTE DISPONIBILITE –

Partie 2: Protocole de redondance du support (MRP)

AVANT-PROPOS

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

La Norme internationale 62439-2 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

Cette deuxième édition annule et remplace la première édition parue en 2010. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- ajout d'une extension de protocole pour sélectionner automatiquement le gestionnaire de redondance du support;
- ajout d'un protocole pour connecter de manière redondante les anneaux du protocole de redondance du support.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65C/834/FDIS	65C/841/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.

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

Cette Norme internationale doit être lue conjointement avec l'IEC 62439-1.

Une liste de toutes les parties de la série IEC 62439, publiées sous le titre général *Réseaux industriels de communication – Réseaux d'automatisme à haute disponibilité*, peut être consultée sur le site web de l'IEC.

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

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

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 ce document en utilisant une imprimante couleur.

INTRODUCTION

La série IEC 62439 précise les principes pertinents relatifs aux réseaux de haute disponibilité qui satisfont aux exigences des réseaux d'automatisation industriels.

A l'état exempt de panne du réseau, les protocoles de la série IEC 62439 assurent une communication de données fiable et conforme à l'ISO/IEC/IEEE 8802-3 (IEEE 802.3) et préservent le caractère déterministe des communications en temps réel. En cas de panne, de retrait et d'insertion d'un composant, ils assurent des temps de reprise déterministes.

Ces protocoles conservent la totalité des fonctions de communication Ethernet classiques utilisées dans le monde professionnel, ce qui permet de continuer à utiliser le logiciel.

Le marché doit disposer de plusieurs solutions réseau, qui présentent des caractéristiques de performance et des capacités fonctionnelles différentes en fonction des différentes exigences d'application. Ces solutions prennent en charge différents mécanismes et topologies de redondance qui sont intégrés à l'IEC 62439-1 et spécifiés dans les autres parties de la série IEC 62439. L'IEC 62439-1 distingue également les différentes solutions en fournissant des lignes directrices aux utilisateurs.

La série IEC 62439 se conforme à la structure et aux termes généraux de la série IEC 61158.

La Commission Electrotechnique Internationale (IEC) attire l'attention sur le fait qu'il est déclaré que la conformité aux dispositions du présent document peut impliquer l'utilisation des droits de propriété intellectuelle relatifs au protocole en anneau donné à l'Article 5. Le Tableau 1 donne une vue d'ensemble des brevets adéquats.

Tableau 1 – Informations de brevet

Numéro	Statut	Pays	Numéro de brevet octroyé ou Numéro d'application (si en attente)	Titre
1	octroyé	US	US 6,430,151	Local networking with redundancy properties having a redundancy manager
	octroyé	CA	CA 2323429	
	octroyé	CN	CN 117195	
	octroyé	NON	NO 330908	
	octroyé	EP (AT, BE, CH, DE, DK, ES, FR, GB, IT, NL, SE)	EP 1062787	

L'IEC 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 assuré à l'IEC qu'il est prêt à négocier des licences avec les demandeurs dans le monde entier, gratuitement ou dans des conditions raisonnables et non discriminatoires. A ce propos, l'énoncé du détenteur des droits de propriété est enregistré à l'IEC. Des informations peuvent être demandées à:

Siemens Aktiengesellschaft
Otto-Hahn-Ring 6
Munich 81739
Allemagne

et

Hirschmann Automation and Control GmbH
Stuttgarter Strasse 45-51
Neckartenzlingen 72654
Allemagne

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. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

L'ISO (www.iso.org/patents) et l'IEC (<http://patents.iec.ch/>) maintiennent à disposition des bases de données en ligne des brevets relatifs à leurs normes. Les utilisateurs sont encouragés à consulter ces bases de données pour obtenir l'information la plus récente sur les droits de propriété.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – RÉSEAUX D'AUTOMATISME A HAUTE DISPONIBILITE –

Partie 2: Protocole de redondance du support (MRP)

1 Domaine d'application

La série IEC 62439 concerne les réseaux de haute disponibilité pour l'automation reposant sur la technologie ISO/IEC/IEEE 8802-3 (IEEE 802.3) (Ethernet).

La présente partie de la série IEC 62439 spécifie un protocole de reprise reposant sur une topologie en anneau, conçu pour réagir de manière déterministe sur une seule défaillance d'une maille interétage ou d'un commutateur du réseau, sous le contrôle d'un nœud du gestionnaire de redondance du support dédié.

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).

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

IEC 61158-6-10:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-10: Spécification du protocole de la couche application – Eléments de type 10*

IEC 61784-1, *Réseaux de communication industriels – Profils – Partie 1: Profils de bus de terrain*

IEC 61784-2, *Réseaux de communication industriels – Profils – Partie 2: Profils de bus de terrain supplémentaires pour les réseaux en temps réel basés sur l'ISO/IEC 8802-3*

IEC 62439-1:2010, *Réseaux industriels de communication – Réseaux d'automatisme à haute disponibilité – Partie 1: Concepts généraux et méthodes de calcul*
IEC 62439-1:2010/AMD1:2012¹

ISO/IEC 10164-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Gestion-systèmes: Fonction de gestion d'objets*

ISO/IEC/IEEE 8802-3, *Standard for Ethernet* (disponible en anglais seulement)

IEEE 802.1D:2004, *IEEE Standard for Local and metropolitan area networks – Media Access Control (MAC) Bridges* (disponible en anglais seulement)

¹ Il existe une version consolidée de cette publication comprenant l'IEC 62439-1:2010 et l'IEC 62439-1:2010/AMD1:2012.

IEEE 802.1Q:2011, *IEEE Standard for Local and metropolitan area networks – Media Access Control (MAC) Bridges and Virtual Bridge Local Area Network* (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 l'IEC 60050-191 et dans l'IEC 62439-1, ainsi que les suivants s'appliquent.

3.1.1

port d'interconnexion

port d'un commutateur utilisé pour interconnecter deux topologies en anneau

3.1.2

topologie d'interconnexion

topologie dans laquelle deux topologies en anneau sont connectées

3.2 Abréviations et acronymes

Pour les besoins du présent document, les abréviations et acronymes donnés dans l'IEC 62439-1 s'appliquent, ainsi que ceux indiqués ci-dessous.

ASE	Application Service Element (élément d'application de service)
CCM	Continuity Check Messages (messages de vérification de la continuité)
CFM	Connectivity Fault Management (gestion de panne de la connectivité)
CFM-MRP	Connectivity Fault Management for MRP (gestion de panne de la connectivité pour MRP)
UC	Unité Centrale
FDB	Filtering Database (base de données de filtrage)
LC-mode	Link Check mode (mode de vérification de liaison)
LSB	Least Significant Bit (bit de poids faible)
MAID	Maintenance Association ID (ID d'association de maintenance)
MD	Maintenance Domain (domaine de maintenance)
MD Level	Maintenance Domain Level (niveau du domaine de maintenance)
MEP	Maintenance End Point (point d'extrémité d'association de maintenance)
MEPID	Maintenance Association End Point ID (ID de point d'extrémité d'association de maintenance)
MIB	Management Information Base (base d'informations de gestion)
MRA	Media Redundancy Automanager (gestionnaire automatique de redondance du support)
MRC	Media Redundancy Client (client de redondance du support)
MRM	Media Redundancy Manager (gestionnaire de redondance du support)
MRP	Media Redundancy Protocol (protocole de redondance du support)
MIC	Media Redundancy Interconnection Client (client d'interconnexion de redondance du support)
MIM	Media Redundancy Interconnection Manager (gestionnaire d'interconnexion de redondance du support)
RC-mode	Ring Check mode (mode de vérification de l'anneau)

3.3 Conventions

Le présent document suit les conventions définies dans l'IEC 62439-1.

4 Vue d'ensemble du MRP

Le présent document présente le protocole MRP (Media Redundancy Protocol, protocole de redondance du support), un protocole de reprise qui repose sur une topologie en anneau, ainsi que l'interconnexion redondante des anneaux du MRP via le protocole d'interconnexion de redondance du support, comme illustré à la Figure 1.

Le MRP est conçu pour réagir de manière déterministe sur une seule défaillance d'une maille interétage ou d'un commutateur de la topologie en anneau ou de la topologie d'interconnexion.

MRP repose sur les fonctions de l'ISO/IEC/IEEE 8802-3 (IEEE 802.3) et de l'IEEE 802.1Q, y compris la base de données de filtrage (FDB, Filtering Data Base); il se trouve entre la couche de liaison de données et la couche d'application (voir Figure 2).

NOTE L'organisation en couches est considérée conforme à l'IEC 61158-1.

Un réseau conforme au MRP doit disposer d'une topologie en anneau à plusieurs nœuds.

L'un des nœuds joue le rôle de gestionnaire de redondance du support (MRM). Le MRM doit observer et contrôler la topologie en anneau afin de réagir en cas de panne du réseau. Le MRM envoie pour cela des trames sur un port en anneau et les reçoit de l'anneau sur un autre port en anneau et inversement.

Les autres nœuds de l'anneau jouent le rôle de clients de redondance du support (MRC). Un MRC réagit en fonction des trames de reconfiguration reçues du MRM; il peut détecter et signaler des changements de liaison sur ses ports en anneau.

Une partie ou l'ensemble des nœuds de l'anneau peut également commencer en tant que gestionnaire automatique de redondance du support (MRA). Les MRA sélectionnent un MRM entre eux via un protocole de sélection. Les autres MRA prennent le rôle de MRC.

Chaque nœud de l'anneau peut détecter la défaillance ou la reprise d'une maille interétage ou d'un nœud avoisinant.

Pour connecter deux anneaux de MRP de manière redondante, des rôles supplémentaires sont affectés à deux nœuds de chaque anneau.

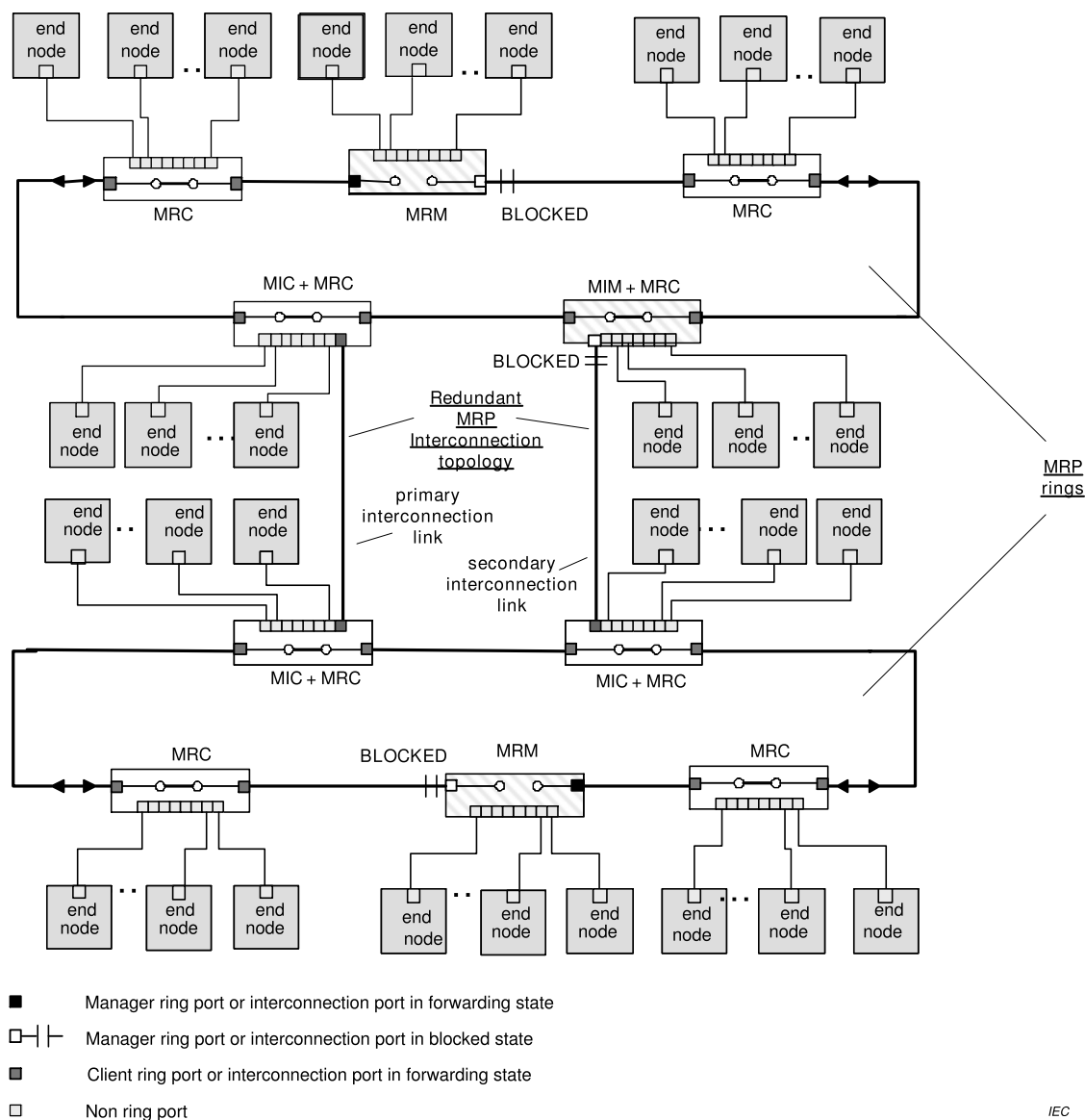
L'un des nœuds joue le rôle de gestionnaire d'interconnexion de redondance du support (MIM), en plus du rôle de MRC ou de MRM. Le MIM doit observer et contrôler la topologie d'interconnexion redondante afin de réagir en cas de panne de l'interconnexion. Pour couvrir un maximum d'applications, deux méthodes de détection sont fournies dans la présente Norme internationale. Pour observer la topologie d'interconnexion, le MIM peut réagir directement aux messages de notification de changement de la liaison du port d'interconnexion (LC-mode) ou envoyer des trames d'essai sur le port d'interconnexion via les anneaux connectés et les recevoir sur ses ports en anneau et inversement (RC-mode).

Les trois autres nœuds de la topologie d'interconnexion jouent le rôle de clients d'interconnexion de redondance du support (MIC), en plus du rôle de MRC ou de MRM. Les MIC réagissent aux trames de reconfiguration reçues par le MIM. Ils peuvent détecter et signaler des changements de liaisons sur leur port d'interconnexion et peuvent publier des messages de notification de changement de liaison.

Comme le décrit la présente Norme internationale, le LC-mode a l'avantage de restreindre la charge de trame d'essai d'interconnexion uniquement au niveau des liaisons d'interconnexion,

alors que le RC-mode a l'avantage de décharger les MIC du traitement des trames d'essai d'interconnexion. La sélection du mode doit être effectuée conformément aux exigences de l'application et ne relève pas du domaine d'application de la présente Norme internationale.

Le fournisseur peut mettre en œuvre le LC-mode seulement, le RC-mode seulement ou les deux à la fois. Le MIM et tous les MIC d'une topologie d'interconnexion de MRP doivent être configurés sur le même mode.



Anglais	Français
End node	Nœud d'extrémité
MRP rings	Anneaux du MRP
Redundant MRP interconnection topology	Topologie d'interconnexion redondante du MRP
Primary interconnection link	Liaison d'interconnexion principale
Secondary interconnection link	Liaison d'interconnexion secondaire
Manager ring port or interconnection port in forwarding state	Port en anneau ou port d'interconnexion du gestionnaire à l'état FORWARDING
Manager ring port or interconnection port in blocked state	Port en anneau ou port d'interconnexion du gestionnaire à l'état BLOCKED
Client ring port or interconnection port in forwarding state	Port en anneau ou port d'interconnexion du client à l'état FORWARDING
Non ring port	Port qui n'est pas en anneau

Figure 1 – Deux anneaux de MRP connectés de manière redondante via l'interconnexion de MRP

Chaque nœud conforme au MRP exige un élément commutateur conforme à l'IEEE 802.1, équipé de deux ports en anneau connectés à un anneau et doit pouvoir tenir au moins l'un des rôles suivants:

- gestionnaire de redondance du support (MRM); ou
- client de redondance du support (MRC); ou
- MRM ou MRC (mais les deux rôles ne doivent pas être actifs en même temps sur le même anneau).

Chaque nœud conforme à l'interconnexion de redondance du support exige un élément commutateur conforme à l'IEEE 802.1, équipé de deux ports en anneau connectés à un anneau, ainsi qu'un port d'interconnexion connecté à un autre anneau de MRP; il doit pouvoir tenir au moins l'un des rôles suivants:

- à la fois MRC et gestionnaire d'interconnexion de redondance du support (MIM) (les deux rôles doivent être actifs en même temps); ou
- MRM et MIM (les deux rôles doivent être actifs en même temps); ou
- à la fois MRC et client d'interconnexion de redondance du support (MIC) (les deux rôles doivent être actifs en même temps); ou
- MRM et MIC (les deux rôles doivent être actifs en même temps).

Plus de deux anneaux de MRP peuvent être connectés de manière redondante en appliquant plusieurs fois la topologie d'interconnexion redondante d'un anneau à d'autres. Ainsi, les anneaux contiennent plusieurs nœuds configurés pour être MIM et MIC. Une connexion redondante de plus de deux anneaux peut également être obtenue par deux nœuds d'un anneau contenant plusieurs instances de MIM ou de MIC.

Un nœud conforme au MRP peut mettre en œuvre plusieurs instances de fonctionnalités relatives au MRC, MRM, MRA, MIM ou MIC. Dans ce cas, il doit respecter les règles de configuration suivantes:

- si au moins une instance de MIM ou MIC est active sur un nœud, alors une seule instance de MRC, MRM ou MRA doit être active sur ce nœud au même moment;
- si plusieurs instances de MRC, MRM ou MRA sont actives sur un nœud, alors aucun MIC ou MIM ne doit être actif sur ce nœud au même moment;
- si plusieurs instances de MRC, MRM ou MRA sont actives sur un nœud, connectant ainsi les anneaux relatifs connectés aux ports en anneau de ce nœud, alors aucun autre nœud dans le réseau ne doit connecter les mêmes anneaux sur ses instances de MRC, MRM ou MRA.

Les définitions et chapitres suivants comportant les définitions du MRP et d'interconnexion de MRP s'appliquent toujours à une instance de MRP ou à une instance d'interconnexion de MRP par nœud conforme. L'activation multiple d'instances de MRP ou d'interconnexion de MRP, conformément aux règles de configuration susmentionnées, pourrait nécessiter des exigences supplémentaires relatives au bon déroulement du traitement de trame et du transfert de trame, qui ne relèvent pas du domaine d'application du présent document.

NOTE Des ports d'interconnexion supplémentaires peuvent être utilisés pour se connecter à des anneaux supplémentaires.

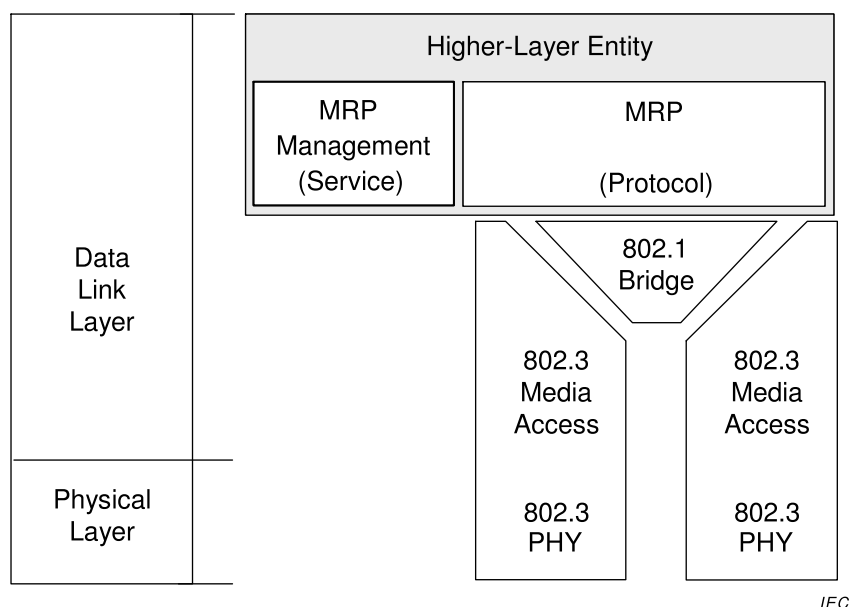
Le MRP est composé d'une entité de service et d'une entité de protocole; voir le modèle en pile à la Figure 2.

L'entité de service spécifie, de manière abstraite, le service visible en externe fourni par la couche liaison de données en termes:

- d'actions et d'événements primitifs du service;
- de paramètres associés à chaque action et événement primitif, ainsi que la forme qu'ils prennent et;
- d'interaction entre ces actions et événements et leurs séquences valides.

Le MRP définit les services fournis:

- à la couche application à la frontière entre la couche application et la couche liaison de données; et
- à la gestion du MRP à la frontière entre la couche liaison de données et la gestion du MRP.



IEC

Anglais	Français
Higher-Layer Entity	Entité de couche supérieure
MRP Management (Service)	Gestion du MRP (Service)
MRP (Protocol)	MRP (Protocole)
Data Link Layer	Couche liaison de données
Physical Layer	Couche physique
Media Access	Accès au support
Bridge	Pont

Figure 2 – MRP en pile

5 Comportement de redondance du support de MRP

5.1 Généralités

Les définitions suivantes relatives aux rôles du protocole de redondance du support font toujours référence à une instance du rôle du MRP.

5.2 Ports en anneau

Le MRM et le MRC doivent être dotés de deux ports en anneau.

Le MRM et le MRC doivent pouvoir détecter la défaillance ou la reprise d'une liaison sur un port en anneau selon des mécanismes reposant sur l'ISO/IEC/IEEE 8802-3 (IEEE 802.3).

Le MRM et le MRC ne doivent pas transférer les trames MRP_Test, MRP_TopologyChange et MRP_LinkChange vers des ports qui ne sont pas en anneau.

NOTE Il est attendu que les ports en anneau se comportent comme si les protocoles STP, RSTP ou MSTP étaient désactivés.

Un port en anneau doit être à l'un des états de port ci-dessous:

– DISABLED

Toutes les trames doivent être abandonnées.

– BLOCKED

Toutes les trames doivent être abandonnées, à l'exception des trames suivantes:

- MRP_TopologyChange, MRP_Test et MRP_TestMgrNAck;
- MRP_LinkChange et MRP_TestPropagate à partir d'un MRC;
- MRP_Option;
- MRP_InTopologyChange et MRP_InLinkStatusPoll à partir d'un MIM;
- MRP_InLinkChange à partir d'un MIC;
- trames spécifiées dans le Tableau 7-10 de l'IEEE 802.1D-2004 pour mettre les ports à l'état "Discarding" (ex: LLDP, IEEE 1588-2008 PTP, trames peer-to-peer);
- trames uniquement générées ou utilisées par les entités de la couche supérieure de ce nœud et qui ne sont jamais transférées.

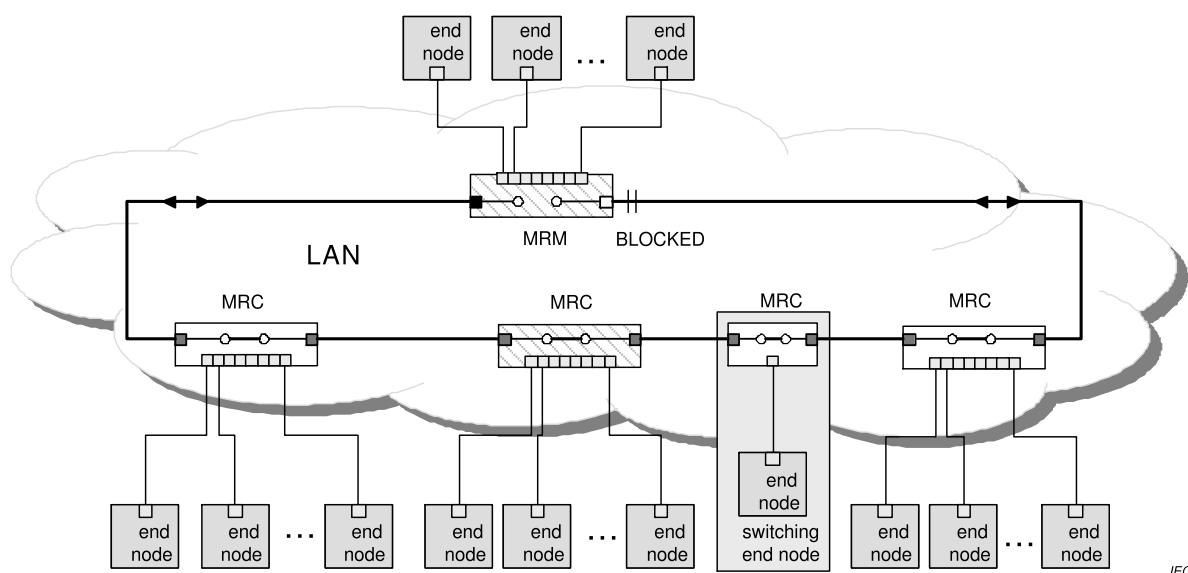
– FORWARDING

Toutes les trames doivent être transférées conformément au comportement relatif au transfert de l'IEEE 802.1D.

NOTE L'IEEE 802.1D fait référence à l'état du port correspondant à BLOCKED en tant que "Discarding".

5.3 Gestionnaire de redondance du support (MRM)

Un port en anneau du MRM doit être connecté à un port en anneau d'un MRC. L'autre port en anneau de ce MRC doit être connecté à un port en anneau de l'autre MRC ou au deuxième port en anneau du MRM, formant ainsi une topologie en anneau comme indiqué à la Figure 3.

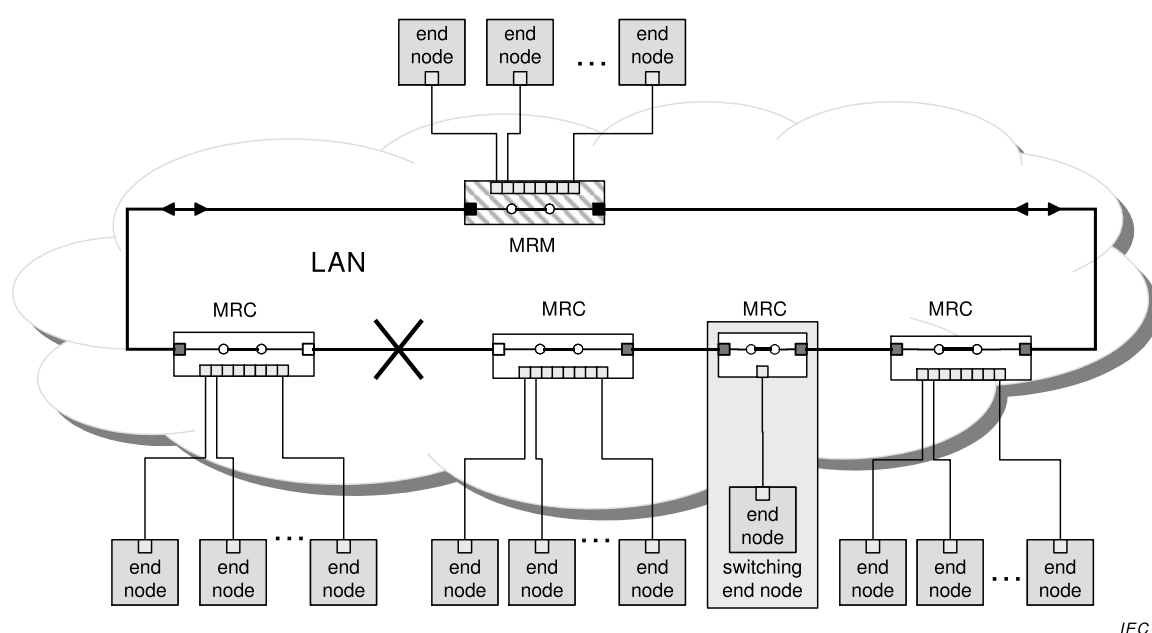


Anglais	Français
end node	nœud d'extrémité
switching end node	nœud d'extrémité de commutation
BLOCKED	BLOQUE

Figure 3 – Topologie en anneau du MRP avec un gestionnaire et des clients

Le MRM doit contrôler l'état de l'anneau en:

- envoyant des trames MRP_Test à intervalle configuré dans les deux directions de l'anneau;
- configurant un port en anneau à l'état FORWARDING et l'autre port en anneau à l'état BLOCKED s'il reçoit ses propres trames MRP_Test (cela signifie que l'anneau est fermé, voir Figure 3);
- configurant les deux ports en anneau à l'état FORWARDING s'il ne reçoit pas ses propres trames MRP_Test sur les deux ports en anneau dans un délai configuré conformément à MRP_TSTdefaultT, MRP_TSTshortT et MRP_TSTNRmax au Tableau 59 (cela signifie que l'anneau est ouvert, voir Figure 4).



IEC

Anglais	Français
end node	nœud d'extrémité
switching end node	nœud d'extrémité de commutation

Figure 4 – Anneau ouvert du MRP avec le MRM

Le mécanisme suivant prend en charge la synchronisation entre le MRM et le MRC dans les changements de topologie en anneau.

Le MRM doit indiquer les changements d'état de l'anneau aux MRC au moyen de trames MRP_TopologyChange.

Le MRM ne doit pas transférer de trames spécifiques au MRP (trames MRP_Test, MRP_TopologyChange, MRP_LinkChange) entre ses ports en anneau.

Si le MRM reçoit une trame MRP_LinkUp ou MRP_LinkDown, il doit réduire son temps de surveillance d'essai conformément au Tableau 59 afin d'accélérer la détection de l'anneau ouvert. Une fois l'anneau ouvert détecté, le MRM doit configurer un port en anneau à l'état FORWARDING et doit envoyer les trames MRP_TopologyChange par ses deux ports en anneau.

Si le paramètre MRP_REACT_ON_LINK_CHANGE est configuré, voir Tableau 41, à la réception d'une trame MRP_LinkDown, le MRM doit immédiatement configurer un port en anneau à l'état FORWARDING puis doit envoyer les trames MRP_TopologyChange par ses ports en anneau.

Le MRM doit envoyer aux MRC une trame MRP_TopologyChange avec le délai, au-delà duquel le changement de topologie en anneau sera appliqué. Le paramètre qui achemine ce délai est appelé MRP_Interval. A l'expiration de ce délai, tous les MRC doivent supprimer les informations sur leurs ports en anneau contenues dans leur base de données de filtrage (FDB).

Chaque MRC doit envoyer le délai configuré dans le paramètre MRP_Interval au MRM dans les trames MRP_LinkUp et MRP_LinkDown pour l'informer du délai dans lequel le MRC changera l'état de son port de BLOCKED à FORWARDING (trame MRP_LinkUp) ou de FORWARDING à BLOCKED (trame MRP_LinkDown).

Les mesures doivent être incluses afin d'éviter que le MRM ne reste à l'état fermé en cas de défaillance du nœud.

Le MRM ne doit pas transférer les trames MRP_InTest, MRP_InLinkChange, MRP_InLinkStatusPoll et MRP_InTopologyChange entre ses ports en anneau si l'un de ses ports en anneau correspond à l'état de port BLOCKED.

Le MRM doit traiter les trames MRP_InTopologyChange. Il doit annuler sa FDB filtrage si une trame MRP_InTopologyChange l'exige dans un intervalle donné MRP_Interval. Il doit envoyer une trame MRP_TopologyChange si une trame MRP_InTopologyChange l'exige afin de garantir que tous les MRC, tous les MIM et tous les MIC dans le domaine de l'anneau annulent leur FDB en même temps.

5.4 Client de redondance du support (MRC)

Chaque MRC doit transférer les trames MRP_Test reçues sur un port en anneau à l'autre port en anneau et inversement.

Si le MRC détecte une défaillance ou une reprise de la liaison d'un port en anneau, il peut éventuellement signaler le changement en envoyant des trames MRP_LinkChange par ses deux ports en anneau. Chaque MRC doit transférer les trames MRP_LinkChange reçues sur un port en anneau à l'autre port en anneau et inversement.

Chaque MRC doit transférer les trames MRP_TopologyChange reçues sur un port en anneau à l'autre port en anneau et inversement. Chaque MRC doit traiter ces trames. Il doit annuler sa FDB si une trame MRP_TopologyChange l'exige dans un intervalle donné MRP_Interval.

Chaque MRC doit transférer les trames MRP_InTest, MRP_InLinkChange, MRP_InTopologyChange et MRP_InLinkStatusPoll entre ses ports en anneau si aucun MIM ni MIC n'est actif sur le nœud au même moment.

5.5 Domaine de redondance

Le domaine de redondance représente un anneau. Par défaut, tous les MRM et MRC appartiennent au domaine par défaut. Un ID de domaine unique peut être affecté sous forme d'attribut clé à l'ensemble des nœuds d'un anneau. Si un appareil appartient à plusieurs anneaux, il est fondamental de différencier les trames en mettant en œuvre plusieurs instances de MRM ou de MRC. Un nœud doit affecter exactement deux ports en anneau uniques par domaine de redondance.

NOTE Un appareil peut disposer de ports autres que les deux attribués au MRP. Ces autres ports ne sont pas influencés par le MRP.

5.6 Vérification de liaison de support

Pour améliorer la qualité de la détection de liaison de certains ports en anneau ou d'interconnexion, un protocole de vérification de liaison de support peut être utilisé. Il doit être utilisé pour le LC-mode.

Le protocole de vérification de la continuité, défini dans la suite de protocole de gestion de panne de la connectivité (CFM) dans l'IEEE 802.1Q-2011 et spécifié ci-après dans le présent document sous le nom gestion de panne de la connectivité pour MRP (CFM-MRP), doit être pris en charge sous forme de protocole de vérification de liaison du support.

Le CFM-MRP doit être utilisé au niveau des ports d'interconnexion du MIM et des MIC s'il exploite l'interconnexion en anneau en LC-mode. Le CFM-MRP peut être utilisé au niveau des ports d'interconnexion s'il exploite l'interconnexion en anneau en RC-mode.

Le CFM-MRP peut également être utilisé au niveau des ports en anneau des appareils du MRP.

5.7 Application du protocole de vérification de la continuité

5.7.1 Généralités

Lors de l'application du protocole CFM-MRP conformément à la spécification du protocole de vérification de la continuité, comme défini dans l'IEEE 802.1Q-2011, pour vérifier les liaisons entre les ports en anneau ou d'interconnexion, ces derniers représentent les points d'extrémité de maintenance (MEP) appartenant au domaine de maintenance (MD) et à une association de maintenance (MA) au sens de l'IEEE 802.1Q-2011.

Un appareil de MRP, qui applique le protocole CFM-MRP à ses ports d'interconnexion ou en anneau, doit créer en interne un événement MAUTypeChangeInd lorsque l'instance destinataire de vérification de la continuité sur son port de MRP a détecté une erreur de point d'extrémité de maintenance à distance (MEP), via la détection d'une violation relative à la durée de vie des messages de vérification de la continuité (CCM) à partir de l'initiateur du CFM-MRP des MEP des appareils de MRP à distance liés.

5.7.2 Intervalle de message de vérification de la continuité

L'intervalle des CCM à utiliser doit être de 3,33 ms ou 10 ms conformément à l'IEEE 802.1Q-2011.

5.7.3 Niveau du domaine de maintenance

Le niveau du domaine de maintenance (MD Level) à utiliser entre deux ports en anneau ou d'interconnexion de deux appareils de MRP doit être de niveau 0 pour les connexions physiques de niveau les plus basses. Par conséquent, l'adresse MAC multidiffusion de CCM fixe 01-80-C2-00-00-30 est affectée, conformément à l'IEEE 802.1Q-2011.

5.7.4 ID d'association de maintenance (MAID, Maintenance Association ID)

L'ID d'association de maintenance (MAID) à utiliser, conformément à l'IEEE 802.1Q-2011, peut être spécifié, comme défini dans le Tableau 2.

Tableau 2 – Codage de l'ID d'association de maintenance (MAID)

Sous-trame	Valeur
Format du nom de domaine de maintenance (Chaîne de caractères)	4
Longueur du nom de domaine de maintenance	Variable
Nom de domaine de maintenance	Chaîne de caractères comportant: – pour les ports en anneau: le nom de domaine du MRP – pour les ports d'interconnexion: Le nom d'interconnexion et une extension de chaîne .prm pour la liaison principale, .sec pour la liaison secondaire ou .na au cours du démarrage.^a
Format court du nom de MA (nombre entier de 2 octets)	3
Courte longueur du nom de MA	2
Court nom de MA	deux octets avec: – pour les ports en anneau: 0x0000 – pour les ports d'interconnexion: ID d'interconnexion MRP_InID.
^a Un initiateur de MEP de CCM sait qu'il effectue l'envoi sur une liaison secondaire si l'initiateur de MEP qui réside sur un MIM ou sur un destinataire de MEP reçoit les trames du CCM d'un initiateur de MEP avec un rôle de MIM spécifié dans la TLV spécifique de l'IEC. Si ce n'est pas le cas, il l'envoie sur une liaison principale.	

5.7.5 ID de point d'extrémité d'association de maintenance (MEPID)

Le contenu de l'ID de point d'extrémité d'association de maintenance de 2 octets ne relève pas du domaine d'application du présent document.

5.7.6 TLV d'ID d'émetteur

L'utilisation de la TLV d'ID d'émetteur conformément à l'IEEE 802.1Q est facultative pour le MRP.

5.7.7 TLV d'état de port

L'utilisation de la TLV d'état de port conformément à l'IEEE 802.1Q est facultative pour le MRP.

5.7.8 TLV d'état d'interface

L'utilisation de la TLV d'état d'interface conformément à l'IEEE 802.1Q est facultative pour le MRP.

5.8 Utilisation avec des diagnostics et des alarmes

Si la valeur TRUE a été affectée à l'attribut Vérification de la redondance du support, les événements de redondance du support doivent déclencher des événements de diagnostic et des notifications d'alarme.

5.9 Diagnostic de l'anneau

Dans un domaine de redondance, le traitement des événements de diagnostic suivants doit être mis en œuvre par chaque MRM:

- Si un appareil est configuré en tant que MRM, mais s'il ne joue pas son rôle de gestionnaire, il doit signaler un événement de diagnostic `MANAGER_ROLE_FAIL` et suspendre le signalement de tous les autres événements de diagnostic de redondance du support qui ont eu lieu pendant qu'il ne jouait pas le rôle de gestionnaire.
- Si un appareil joue le rôle de gestionnaire et s'il détecte un autre MRM actif, il doit signaler l'événement `MULTIPLE_MANAGERS`. Cet événement peut se produire simultanément avec l'événement d'état `RING_OPEN` de l'anneau.
- Si un appareil joue le rôle de gestionnaire et s'il détecte un anneau ouvert, il doit signaler l'événement `RING_OPEN`.

Ces événements doivent être signalés à l'aide du service State Change (voir 7.3).

NOTE La présence de trames `MRP_Test` permet de vérifier l'existence d'un MRM.

5.10 Plusieurs MRM dans un seul anneau: Option de vote du gestionnaire

5.10.1 Généralités

L'anneau ne doit comporter qu'un seul MRM actif, même si plusieurs nœuds ont cette capacité.

NOTE 1 La présence de plusieurs MRM actifs amène l'anneau à se diviser lui-même en plusieurs segments.

Comme alternative à la configuration manuelle du rôle de MRP, si plusieurs nœuds ont la capacité de devenir un MRM dans l'anneau, une option de sélection du gestionnaire améliorée spécifiée dans la présente Norme internationale peut être utilisée pour décider lequel d'entre eux devra finalement devenir le MRM, alors que les autres nœuds joueront le rôle de MRC comme indiqué à la Figure 5 et la Figure 6. A cet égard, les nœuds avec la capacité de jouer le rôle de MRM font l'objet de priorités différentes qui doivent être acheminées dans la zone `MRP_Prio` de la trame `MRP_Test`.

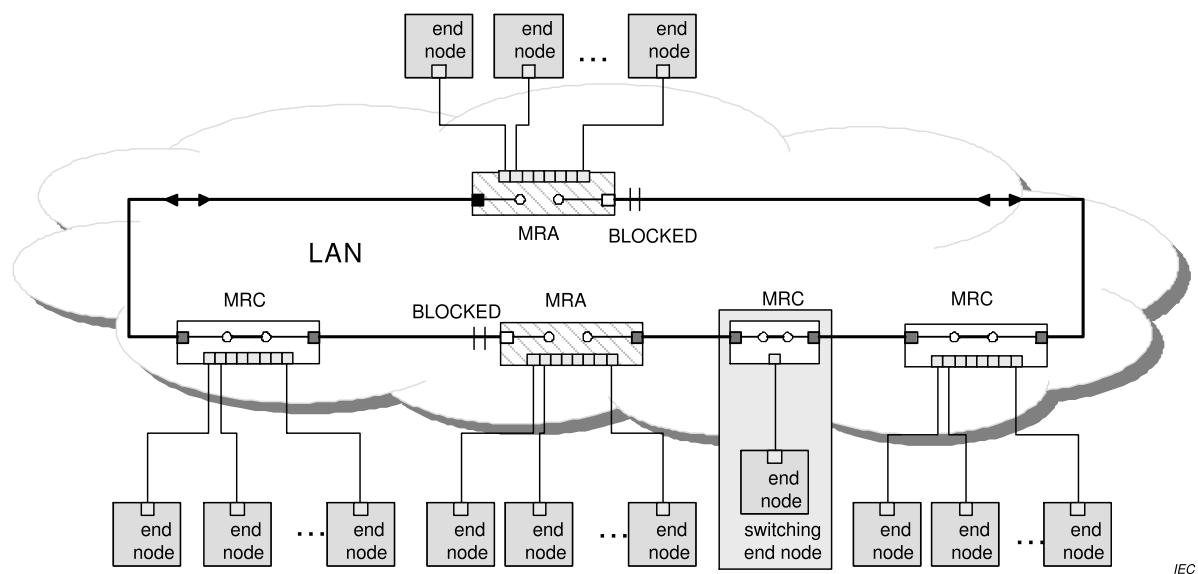
Si une option destinée à plusieurs MRM d'un seul anneau est utilisée, alors tous les MRM de cet anneau doivent prendre en charge la même option de sélection du gestionnaire, comme défini dans le présent document.

Les MRM qui prennent en charge l'option de sélection du gestionnaire sont appelés gestionnaires automatiques de redondance du support ou MRA.

Le rôle des MRA est un rôle de MRP qui comprend à la fois les rôles MRM de type MRP et MRC. En outre, le MRA a la capacité de passer à tout moment, en fonction du protocole de vote, de son rôle opérationnel initial de gestionnaire (MRA) au rôle opérationnel client (MRC), et vice-versa, en coordination avec d'éventuels autres MRA dans l'anneau.

NOTE 2 Pour souligner ce principe, les explications suivantes utilisent l'expression MRM pour le MRA restant qui assure la gestion de l'anneau, bien que le dispositif continue d'être MRA, et donc toujours capable de voter à nouveau.

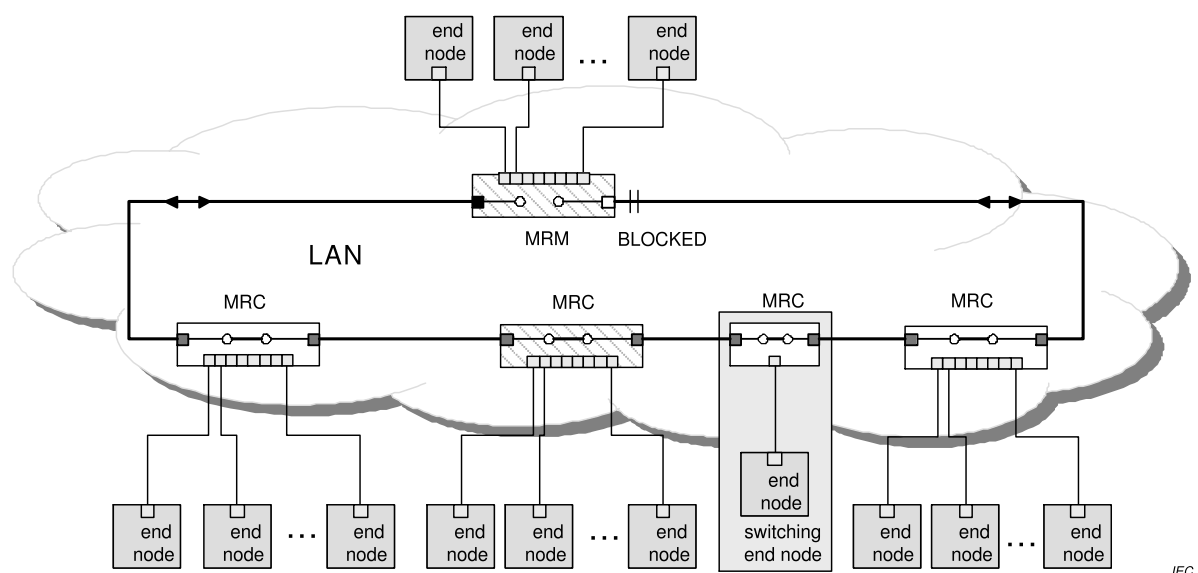
La Figure 5 présente un anneau de MRP au démarrage du réseau avec deux appareils qui jouent le rôle administratif de MRA.



Anglais	Français
end node	nœud d'extrémité
switching end node	nœud d'extrémité de commutation
BLOCKED	BLOQUE

Figure 5 – Anneau de MRP avec un MRA au démarrage du réseau

La Figure 6 présente le même anneau de réseau après la mise en place du processus de sélection du gestionnaire. Le MRA est passé aux rôles de MRM et MRC du MRP.



Anglais	Français
end node	nœud d'extrémité
switching end node	nœud d'extrémité de commutation
BLOCKED	BLOQUE

Figure 6 – Anneau de MRP après le processus de sélection du gestionnaire

5.10.2 Principe de base du processus de sélection du gestionnaire

La sélection du gestionnaire est un processus facultatif uniquement destiné aux appareils qui peuvent jouer le rôle de MRM dans le MRP.

La phase de sélection du gestionnaire du MRP doit se dérouler juste après le démarrage de l'appareil, mais avant que l'appareil ne commence à prendre l'un des éventuels rôles de MRM ou de MRC du MRP.

La façon dont la fonctionnalité de base du processus de sélection du gestionnaire doit se dérouler est décrite ci-après:

- Chaque nœud de réseau de l'anneau avec les capacités de tenir le rôle de MRM et pris en charge par l'option de sélection du gestionnaire, appelé alors gestionnaire (automatique) ou simplement MRA, peut être défini pour le rôle de gestionnaire (automatique) du MRP aux valeurs d'usine par défaut.
- Chaque MRA doit se voir affecter une valeur de priorité (MRP_Prio). Enfin, cette valeur MRP_Prio, associée à l'adresse MAC, aboutit sur une priorité unique. Plus la valeur MRP_Prio est faible, plus la priorité est élevée. Si les appareils ont des valeurs MRP_Prio identiques, la valeur de l'adresse MAC de l'appareil tranche. Plus l'adresse MAC est faible, plus la priorité est élevée.
- Au démarrage, chaque nœud de réseau de MRA doit lancer le processus de sélection du gestionnaire et doit envoyer les trames MRP_Test.
- Les autres MRA doivent recevoir les trames MRP_Test et doivent comparer leur propre priorité avec la priorité du prochain MRA. S'ils ont bien la priorité la plus élevée, ils doivent obliger le gestionnaire à distance qui envoie la trame MRP_Test à abandonner le rôle MRA pour prendre le rôle de MRC via l'envoi d'un accusé de réception de gestionnaire d'essai négatif.
- Le MRA ou le MRM dont la priorité est la plus élevée doit devenir l'unique MRM de l'anneau. Il doit dès lors commencer à gérer l'anneau conformément au MRP.
- Si le MRM choisi à l'aide de l'option de sélection du gestionnaire n'appartient plus à l'anneau ou fait preuve de défaillances pendant l'exploitation, le processus de sélection doit recommencer et la prochaine priorité de MRA doit prendre le relais.
- Si un MRA supplémentaire est intégré à un anneau comptant déjà un MRM sélectionné, alors ce dernier échangera les trames MRP_Test avec le MRA récemment intégré afin de décider lequel des deux restera ou deviendra le MRM.
- Pour que l'option de sélection du gestionnaire fonctionne correctement, tous les appareils de l'anneau dotés du rôle de gestionnaire doivent être définis comme gestionnaires (automatiques) du MRP ou doivent conserver le rôle de clients du MRP. En revanche, au moins un appareil de l'anneau doit être défini comme gestionnaire (automatique) ou gestionnaire du MRP.
- Si un MRM qui ne dispose pas des capacités suffisantes pour devenir un MRA fait partie de l'anneau de réseau, indépendamment du fait que l'anneau soit ouvert ou fermé, tous les MRA doivent être configurés pour le rôle de MRC.

NOTE La configuration du MRA sur le rôle du MRC n'est pas automatiquement exécutée par le processus de vote du gestionnaire automatique, mais bien par l'utilisateur.

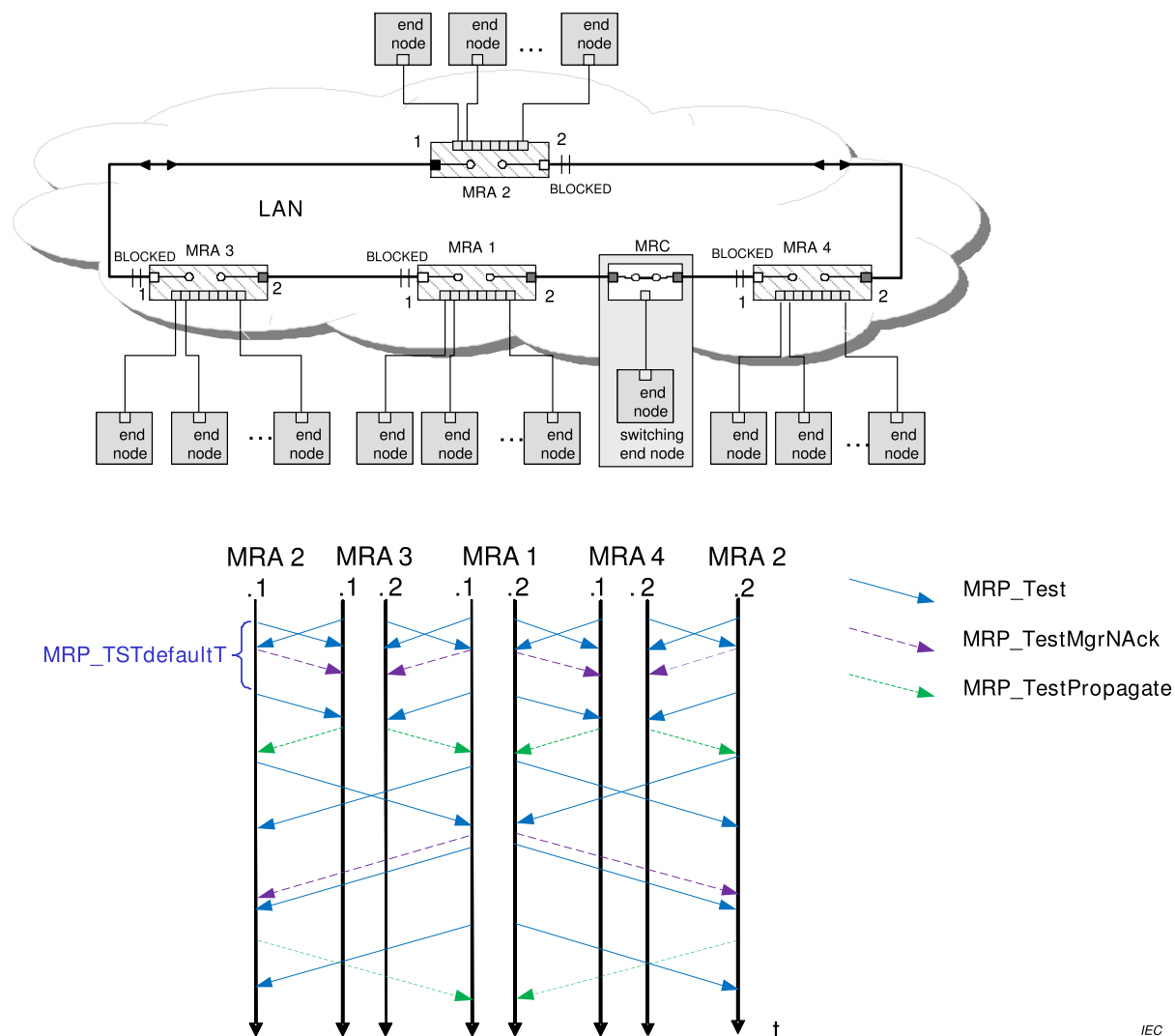
5.10.3 Le processus de sélection du gestionnaire

Le processus de sélection du gestionnaire doit se dérouler de la manière décrite ci-après:

- Au démarrage, tous les MRA doivent lancer le processus de sélection du gestionnaire.
- Chaque MRA doit commencer à envoyer les trames MRP_Test sur les deux ports en anneau. La trame MRP_Test doit contenir la valeur MRP_Prio de priorité des MRA.
- Les trames MRP_Test reçues ne doivent être transférées qu'à l'hôte local.

- Les valeurs de priorité des gestionnaires à distance contenues dans les trames MRP_Test reçues doivent être comparées par les MRA avec leur priorité, comme décrit en 5.10.2. Si la priorité des MRA est supérieure à la priorité reçue, une trame d'accusé de réception de gestionnaire d'essai négatif (MRP_TestMgrNAck) doit être envoyée, accompagnée de l'adresse MAC des gestionnaires à distance en guise d'informations supplémentaires.
- Si un accusé de réception de gestionnaire d'essai négatif (MRP_TestMgrNAck) avec sa propre adresse MAC est reçu, le MRA destinataire doit initier sa transition vers le rôle du client (MRC); pour cela:
 - il enregistre l'adresse MAC et la priorité du gestionnaire dont la priorité est la plus élevée;
 - il joue le rôle de client et diffuse le changement de rôle, ainsi que les informations relatives au gestionnaire dont la priorité est la plus élevée, via une trame MRP_TestPropagate envoyée sur ses deux ports en anneau.
- Lorsqu'il endosse le rôle de client de MRP, il doit transférer les trames MRP_Test entre les ports en anneau, comme un MRC normal. Il doit également transférer les trames MRP_Test à l'UC de l'hôte.
- Le MRA qui joue le rôle de client doit ensuite surveiller si le gestionnaire enregistré dont la priorité est la plus élevée fonctionne toujours, à savoir s'il reçoit encore des trames MRP_Test de la part dudit gestionnaire.
- La trame MRP_TestPropagate doit être utilisée pour informer les autres appareils qui correspondent à des MRA jouant le rôle de clients au sujet du changement de rôle et du nouveau gestionnaire dont la priorité est la plus élevée. En fonction de cela, les clients qui reçoivent cette trame doivent mettre à jour les informations qu'ils possèdent sur le gestionnaire dont la priorité est la plus élevée. Ainsi, les clients conservent leur rôle de client en cas de changement du rôle du gestionnaire surveillé dont la priorité est la plus élevée.
- Les trames MRP_TestMgrNAck et MRP_TestPropagate doivent être envoyées par l'intermédiaire de l'adresse multidiffusion de destination MC_TEST.
- Si le MRA qui joue le rôle de client reconnaît (via l'expiration de la surveillance de la réception de la trame MRP_Test) la défaillance du gestionnaire dont la priorité est la plus élevée, il doit reprendre le rôle de gestionnaire. Il doit par conséquent lancer un nouveau processus de sélection du gestionnaire avec les autres MRA qui, pour des raisons semblables, ont repris tout comme lui leur rôle de gestionnaire.

La Figure 7 présente l'évolution des échanges de trame au cours de la phase de sélection du gestionnaire:

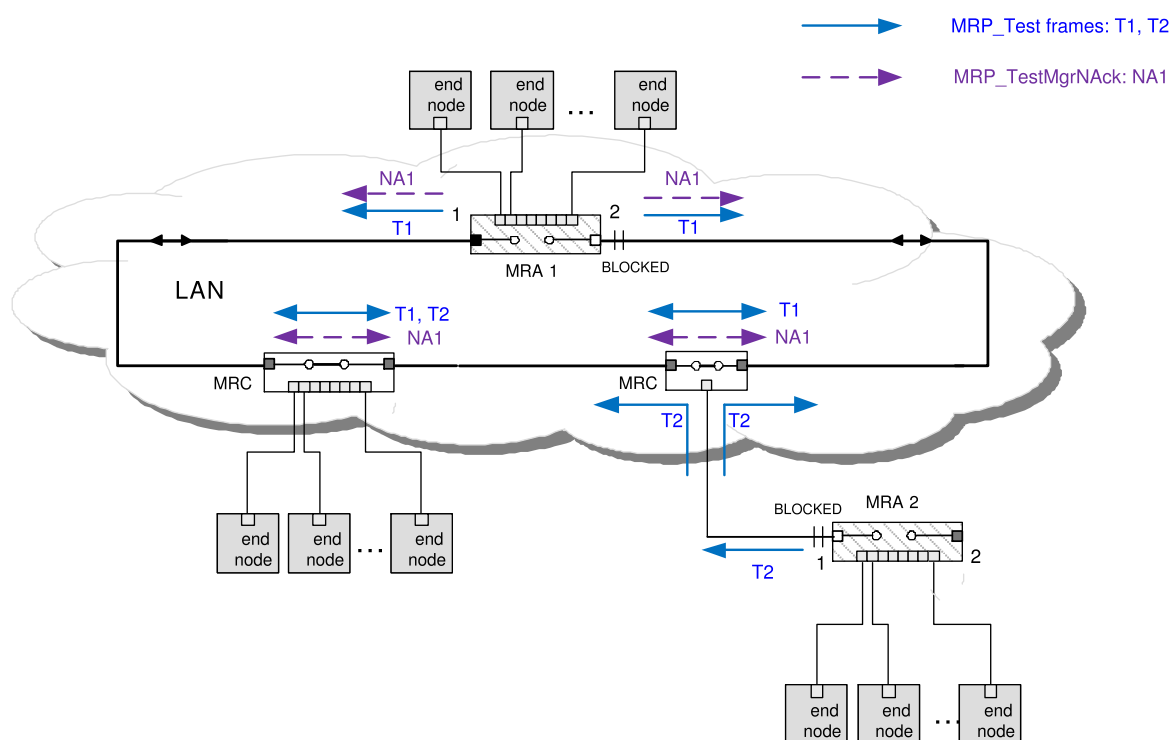


Anglais	Français
end node	nœud d'extrémité
switching end node	nœud d'extrémité de commutation
BLOCKED	BLOQUE

Figure 7 – Processus de sélection du gestionnaire

- Un établissement de liaison (accusé de réception d'essai négatif) est exigé pour garantir que le gestionnaire dont la priorité est la plus élevée se trouve dans l'anneau. Cet établissement de liaison est nécessaire, conformément aux règles de transfert de trame relative au commutateur, car il permet aux trames MRP_Test d'un appareil qui joue le rôle de gestionnaire situé hors de l'anneau, connecté via son port en anneau à un port qui n'est pas en anneau d'un appareil de MRP qui fait partie de l'anneau, d'être également transférées dans l'anneau, si un commutateur sans capacité de filtrage "entrant" ("filtrage entrant") est utilisé.

La Figure 8 illustre un anneau de MRP avec un MRA (MRA2) situé hors de l'anneau. Le MRA2 est connecté via l'un de ses ports en anneau à un port qui n'est pas en anneau d'un appareil qui fait partie de l'anneau. Les trames respectent les règles de transfert définies en 5.2.



Anglais	Français
end node	nœud d'extrémité
switching end node	nœud d'extrémité de commutation
BLOCKED	BLOQUE

Figure 8 – MRA situé hors de l'anneau de MRP

- Même si le gestionnaire MRA1 dont la priorité est plus basse reçoit les trames d'essai de la part du gestionnaire MRA2 dont la priorité est plus élevée, le MRA2 ne reçoit pas les trames d'essai de la part du MRA1, car elles ne sont pas transférées vers des ports qui ne sont pas en anneau, ce qui les exclut du processus de sélection. Le MRA2 n'envoie donc pas d'accusé de réception d'essai négatif et le MRA1 gardera son rôle de MRM. Si le MRA1 était doté d'une priorité plus élevée que le MRA2, le MRA1 enverrait des accusés de réception d'essai négatifs, mais ces derniers ne parviendraient pas jusqu'au MRA2, puisqu'ils ne sont pas transférés non plus vers les ports qui ne sont pas en anneau.

5.11 BLOCKED non pris en charge (option)

Si un MRC ne peut pas prendre en charge l'état de port BLOCKED au niveau de ses ports en anneau, il doit le signaler dans le paramètre correspondant des trames MRP_LinkChange.

Si un MRC ne prend pas en charge l'état BLOCKED dans un anneau, un MRM doit alors prendre en charge des fonctionnalités supplémentaires (voir Tableau 40 et Tableau 41, MRP_BLOCKED_SUPPORTED).

5.12 Port d'interconnexion

Le MIM et le MIC doivent être équipés d'un port d'interconnexion.

Le MIM et le MIC doivent pouvoir détecter la défaillance ou la reprise d'une liaison d'un port d'interconnexion comme suit:

- à l'aide de mécanismes de détection de liaison d'après l'ISO/IEC/IEEE 8802-3 (IEEE 802.3)

En outre, lorsqu'un des mécanismes suivants est appliqué ou les deux:

- protocole de vérification de liaison du support entre les ports d'interconnexion. Le protocole de vérification de liaison du support doit être utilisé au niveau du port d'interconnexion du MIM et des MIC si le MIM est configuré en mode de vérification de liaison (LC-mode) et il peut être utilisé si le MIM est configuré en mode de vérification de l'anneau (RC-mode).
- trame MRP_InTest périodique qui doit être envoyée et reçue par le MIM sur ses ports en anneau et son port d'interconnexion et doit être transférée par les MIC sur leurs ports en anneau et leur port d'interconnexion. Ce mécanisme doit être utilisé pour le RC-mode et ne doit pas être utilisé pour le LC-mode.

Le MIM et le MIC ne doivent pas transférer les trames MRP_InTest, MRP_InTopologyChange, MRP_InLinkStatusPoll et MRP_InLinkChange à d'autres ports que les ports en anneau ou le port d'interconnexion.

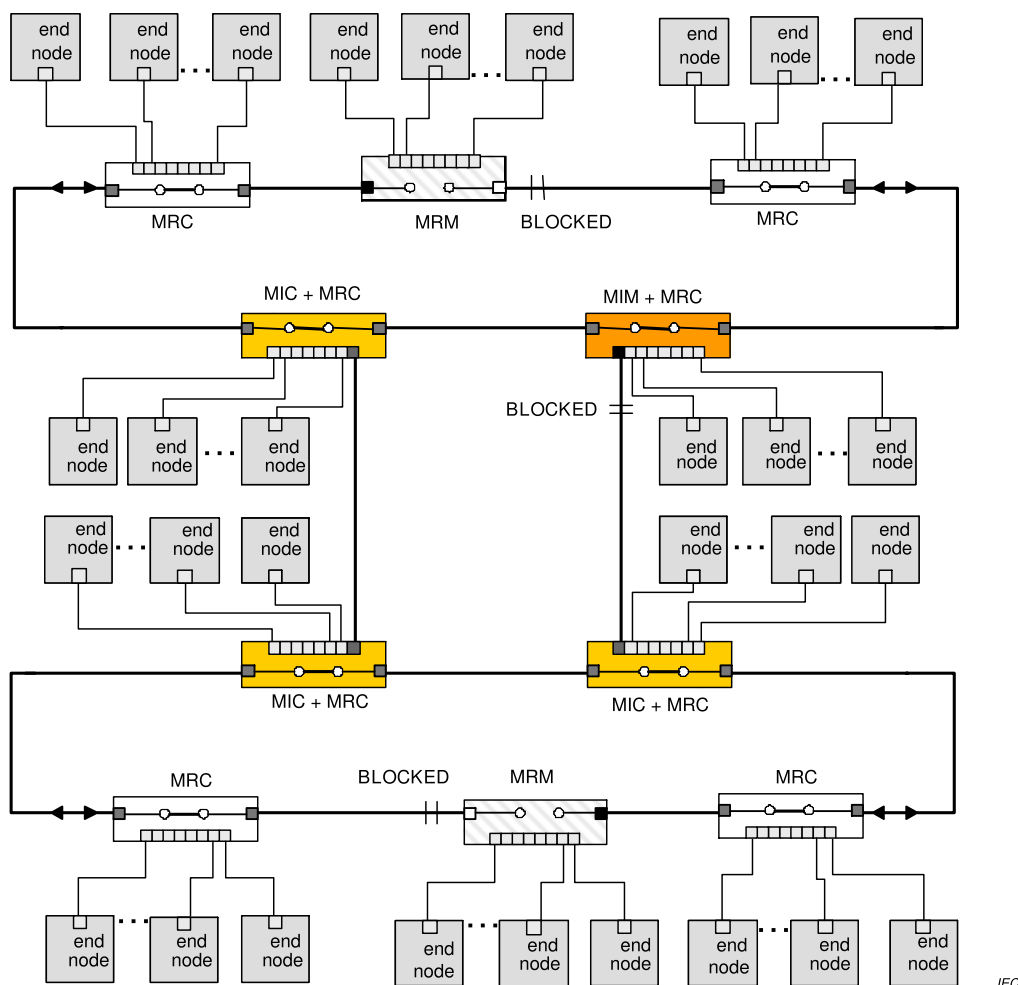
Le port d'interconnexion doit être à l'un des états de port ci-dessous:

- DISABLED
Toutes les trames doivent être abandonnées.
- BLOCKED
Toutes les trames doivent être abandonnées, à l'exception des trames suivantes:
 - MRP_InTest;
 - MRP_InLinkChange et MRP_InTopologyChange;
 - MRP_InLinkStatusPoll;
 - Trames de vérification de liaison du support du CFM-MRP conformément à l'IEEE 802.1Q-2011;
 - Trames spécifiées dans le Tableau 7-10 de l'IEEE 802.1D-2004 pour mettre les ports à l'état "Discarding" (ex: LLDP, IEEE 1588-2008 PTP, trames peer-to-peer);
 - Trames uniquement générées ou utilisées par les entités de la couche supérieure de ce nœud et qui ne sont jamais transférées.
- FORWARDING
Toutes les trames doivent être transférées conformément au comportement relatif au transfert de l'IEEE 802.1D.

NOTE L'IEEE 802.1D fait référence à l'état du port correspondant à BLOCKED en tant que "Discarding".

5.13 Gestionnaire d'interconnexion de redondance du support (MIM)

Le port d'interconnexion du MIM doit être connecté au port d'interconnexion d'un MIC dans un autre anneau du MRP; le port d'interconnexion d'un MIC dans le même anneau doit être connecté au port d'interconnexion d'un MIC dans cet autre anneau du MRP. Une topologie d'interconnexion du MRP est alors formée comme indiqué à la Figure 9.



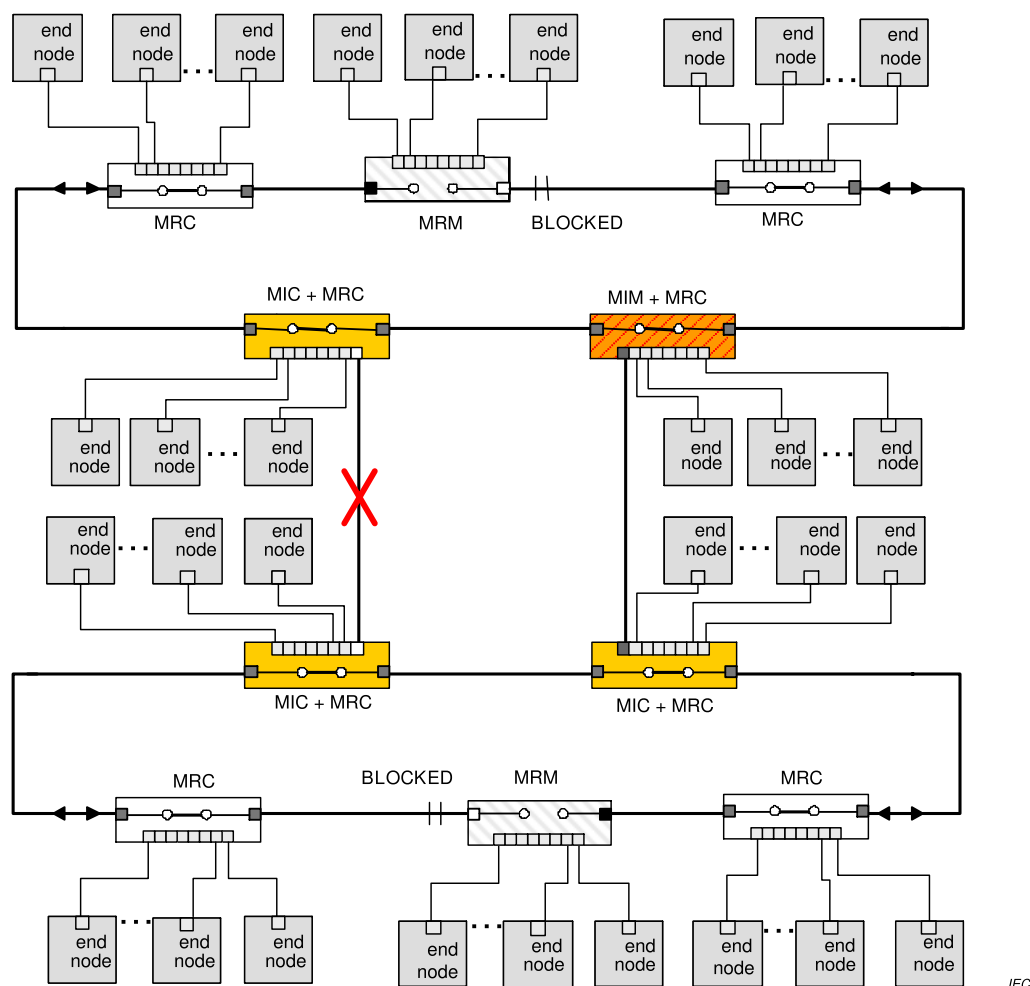
Anglais	Français
end node	nœud d'extrémité
switching end node	nœud d'extrémité de commutation
BLOCKED	BLOQUE

Figure 9 – Topologie d'interconnexion du MRP

Le MIM doit contrôler l'état de son port d'interconnexion comme suit:

- en réagissant directement uniquement aux notifications de changement de la liaison du port d'interconnexion des MIC et à partir de son propre port d'interconnexion (mode de vérification de liaison: LC-mode);
- ou en réagissant de manière facultative aux trames MRP_InTest (mode de vérification de l'anneau: RC-mode);
- en configurant son port d'interconnexion en état BLOCKED si:
 - il est en LC-mode et reçoit des notifications d'établissement de liaison du port d'interconnexion à partir de son propre port d'interconnexion et à partir des MIC de l'interconnexion (cela signifie que la topologie d'interconnexion est fermée, voir Figure 9);
 - il est en RC-mode et reçoit ses propres trames MRP_InTest (cela signifie que la topologie d'interconnexion est fermée, voir Figure 9).
- en configurant le port d'interconnexion en état FORWARDING si:

- il est en LC-mode et reçoit une notification d'interruption de liaison du port d'interconnexion à partir d'au moins un des MIC de l'interconnexion (cela signifie que la topologie d'interconnexion est ouverte, voir Figure 10);
- il est en RC-mode et ne reçoit pas ses propres trames MRP_Test dans un délai configuré conformément à $MRP_IN_TSTdefaultT$ et $MRP_IN_TSTNRmax$ au Tableau 59 (cela signifie que la topologie d'interconnexion est ouverte, voir Figure 10).



Anglais	Français
end node	nœud d'extrémité
switching end node	nœud d'extrémité de commutation
BLOCKED	BLOQUE

Figure 10 – Interconnexion en anneau du MRP ouverte

Le mécanisme suivant prend en charge la synchronisation entre le MIM et le MIC dans les changements de topologie d'interconnexion.

Le MIM doit indiquer les changements dans l'état de topologie d'interconnexion aux MIC, ainsi qu'aux MRM dans les anneaux connectés du MRP, au moyen des trames MRP_InTopologyChange.

Lorsqu'il est en LC-mode, le MIM doit, après le démarrage et la reconnaissance d'un établissement de liaison au niveau du port d'interconnexion, publier une trame MRP_InLinkStatusPoll au niveau de ses ports en anneau, pour interroger l'état d'interconnexion redondante des MIC.

Lorsqu'il est en RC-mode, le MIM doit, après le démarrage et la reconnaissance d'un établissement de liaison au niveau de son port d'interconnexion, envoyer et recevoir de manière cyclique des trames MRP_InTest au niveau de ses ports en anneau et de son port d'interconnexion, pour détecter l'état de la topologie d'interconnexion.

Lorsque la topologie d'interconnexion passe de l'état "ouverte" à l'état "fermée" ou inversement, le MIM doit envoyer les trames MRP_InTopologyChange par ses deux ports en anneau et son port d'interconnexion vers les MIC et les MRM afin de lancer le changement de topologie d'interconnexion.

Si le MIC reçoit une trame MRP_InTopologyChange, alors le MIC doit, après la reconnaissance d'un établissement de liaison au niveau de son port d'interconnexion, modifier l'état de port de son port d'interconnexion en FORWARDING.

Si les MRM reçoivent une trame MRP_InTopologyChange, tous les MRM doivent envoyer des trames MRP_TopologyChange sur les deux ports en anneau, avec le délai. Après cela, tous les MRC et MIC des anneaux connectés du MRP effaceront leur base de données de filtrage (FDB).

Le MIM ne doit pas transférer ses propres trames MRP_InTest entre les ports en anneau et son port d'interconnexion.

Le MIM doit transférer les trames MRP_InLinkChange, MRP_InLinkStatusPoll et MRP_InTopologyChange reçues sur un port en anneau à l'autre port en anneau et inversement. Le MIM doit traiter les trames MRP_InLinkChange et MRP_InTopologyChange. Le MIM ne doit pas transférer les trames MRP_InLinkChange, MRP_InLinkStatusPoll et MRP_InTopologyChange s'il a reçu ces trames au niveau de son port d'interconnexion.

Chaque MIC doit envoyer le délai configuré dans le paramètre MRP_Interval au MIM dans les trames MRP_InLinkUp et MRP_InLinkDown pour l'informer du délai dans lequel le MIC changera l'état de son port d'interconnexion de BLOCKED à FORWARDING (trame MRP_InLinkUp) ou de FORWARDING à BLOCKED (trame MRP_InLinkDown).

Chaque MIM doit prendre en charge l'état de port BLOCKED au niveau du port d'interconnexion.

5.14 Client d'interconnexion de redondance du support (MIC)

Le port d'interconnexion du MIC doit être connecté au port d'interconnexion d'un MIM dans un autre anneau du MRP; le port d'interconnexion d'un autre MIC dans le même anneau doit être connecté au port d'interconnexion d'un MIC dans cet autre anneau du MRP. Une topologie d'interconnexion du MRP est ainsi formée; voir Figure 9.

Chaque MIC doit transférer les trames MRP_InTest reçues sur un port en anneau à l'autre port en anneau et au port d'interconnexion. Chaque MIC doit transférer les trames MRP_InTest reçues sur le port d'interconnexion aux deux ports en anneau. Si le MIC détecte une défaillance ou une reprise de la liaison du port d'interconnexion, il doit signaler le changement en envoyant des trames MRP_InLinkChange par ses deux ports en anneau. Chaque MIC doit transférer les trames MRP_InLinkChange reçues sur un port en anneau à l'autre port en anneau et inversement. Chaque MIC doit transférer les trames MRP_InLinkChange reçues sur l'un des ports en anneau au port d'interconnexion.

Chaque MIC doit transférer les trames MRP_InTopologyChange reçues sur un port en anneau à l'autre port en anneau et inversement. Chaque MIC doit traiter les trames MRP_InTopologyChange. Le MIC doit, après la reconnaissance d'un établissement de liaison au niveau de son port d'interconnexion, modifier l'état de port de son port d'interconnexion en FORWARDING.

Après avoir reçu une trame MRP_InLinkStatusPoll de la part du MIM, chaque MIC doit répondre avec une trame d'état MRP_InLinkChange qui indique l'état de liaison en cours de son port d'interconnexion.

Chaque MIC doit prendre en charge l'état de port BLOCKED au niveau du port d'interconnexion.

5.15 Domaine d'interconnexion

Le domaine d'interconnexion représente une topologie d'interconnexion. Le MIM et les MIC d'une topologie d'interconnexion appartiennent au même domaine d'interconnexion. Un ID d'interconnexion unique doit être affecté sous forme d'attribut clé, surtout si le MIM et les MIC appartiennent à des anneaux dotés de plusieurs interconnexions. Un nœud doit affecter exactement un port d'interconnexion unique par domaine d'interconnexion.

Chaque topologie d'interconnexion du réseau doit disposer d'un ID d'interconnexion unique.

Un seul MIM doit être actif dans un domaine d'interconnexion.

NOTE Il est attendu que les ports d'interconnexion se comportent comme si les protocoles STP, RSTP ou MSTP étaient désactivés.

5.16 Diagnostic d'interconnexion

Dans un domaine d'interconnexion, le traitement des événements de diagnostic suivants doit être mis en œuvre par chaque MIM.

- Si un appareil est configuré en tant que MIM, mais qu'il ne joue pas ce rôle, il doit signaler un événement de diagnostic MRP_INTERCONNECTION_MANAGER_ROLE_FAIL et suspendre le signalement de tous les autres événements de diagnostic du protocole d'interconnexion de redondance du support qui ont eu lieu pendant qu'il ne remplissait pas le rôle de gestionnaire.
- Si un appareil joue le rôle de gestionnaire d'interconnexion du MRP (MIM) et s'il détecte une topologie d'interconnexion ouverte ou s'il est informé de cette dernière, il doit signaler l'événement INTERCONNECTION_OPEN.

Ces événements doivent être signalés à l'aide du service State Change (voir 7.3).

6 Spécification de classe du MRP

6.1 Généralités

L'élément d'application de service (Application Service Element, ASE) du MRP définit deux types d'objets.

6.2 Modèle

6.2.1 Modèle de redondance du support

Un objet de MRP est décrit par le modèle suivant:

ASE: ASE de redondance du support

CLASSE: Redondance du support

ID DE CLASSE: pas utilisé

CLASSE PARENTE: IEEE 802.3, IEEE 802.1D

ATTRIBUTS:

1.	(m)	Attribut clé:	ID de domaine
2.	(m)	Attribut:	Nom de domaine
3.	(m)	Attribut:	ID 1 du port d'anneau
4.	(m)	Attribut:	ID 2 du port d'anneau
5.	(o)	Attribut:	ID VLAN
6.	(m)	Attribut:	Rôle attendu (MANAGER (AUTO), MANAGER, CLIENT)
7.	(c)	Contrainte:	Rôle attendu = MANAGER
7.1	(m)	Attribut:	Priorité du gestionnaire
7.2	(m)	Attribut:	Intervalle de modification de la topologie
7.3	(m)	Attribut:	Nombre de répétitions de modification de la topologie
7.4	(m)	Attribut:	Intervalle d'essai court
7.5	(m)	Attribut:	Intervalle d'essai par défaut
7.6	(m)	Attribut:	Nombre de surveillances d'essai
7.7	(m)	Attribut:	MRC non bloquant pris en charge (TRUE, FALSE)
7.8	(c)	Contrainte:	MRC non bloquant pris en charge = TRUE
7.8.1	(m)	Attribut:	Nombre étendu de surveillances d'essai
7.9	(o)	Attribut:	Modification de liaison de réaction (TRUE, FALSE)
7.10	(m)	Attribut:	Vérification de la redondance du support (TRUE, FALSE)
7.10.1	(c)	Contrainte:	Vérification de la redondance du support = TRUE
7.10.1.1	(m)	Attribut:	Etat réel du rôle
7.10.1.2	(m)	Attribut:	Etat réel de l'anneau
7.10.1.3	(o)	Attribut:	Etat du port 1 du port de l'anneau
7.10.1.4	(o)	Attribut:	Etat du port 2 du port de l'anneau
8.	(c)	Contrainte:	Rôle attendu = CLIENT
8.1	(m)	Attribut:	Intervalle d'arrêt de la liaison
8.2	(m)	Attribut:	Intervalle de fonctionnement de la liaison
8.3	(m)	Attribut:	Nombre de modifications de la liaison
8.4	(o)	Attribut:	Etat du port 1 du port de l'anneau
8.5	(o)	Attribut:	Etat du port 2 du port de l'anneau
8.6	(m)	Attribut:	Etat BLOCKED pris en charge (TRUE, FALSE)

SERVICES:

1	(m)	OpsService:	Démarrage MRM
2	(m)	OpsService:	Arrêt MRM
3	(o)	OpsService:	Modification de l'état
4	(m)	OpsService:	Démarrage MRC
5	(m)	OpsService:	Arrêt MRC
6	(o)	OpsService:	Lecture MRM
7	(o)	OpsService:	Lecture MRC

6.2.2 Modèle d'interconnexion de redondance du support

Un objet d'interconnexion de MRP est décrit par le modèle suivant:

ASE: ASE d'interconnexion de redondance du support

CLASSE: Interconnexion de redondance du support

ID DE CLASSE: pas utilisé

CLASSE PARENTE: IEEE 802.3, IEEE 802.1D

ATTRIBUTS:

1.	(m)	Attribut clé:	ID d'interconnexion
2.	(m)	Attribut:	Nom d'interconnexion
3.	(m)	Attribut:	ID de port d'interconnexion
4.	(o)	Attribut:	Etat de port d'interconnexion
5.	(o)	Attribut:	ID VLAN d'interconnexion
6.	(o)	Attribut:	Mode d'interconnexion
7.	(m)	Attribut:	Rôle d'interconnexion (MANAGER, CLIENT)
8.	(c)	Contrainte:	Rôle d'interconnexion = MANAGER
8.1	(m)	Attribut:	Intervalle de modification de topologie d'interconnexion
8.2	(m)	Attribut:	Nombre de répétitions de modification de topologie d'interconnexion
8.3	(o)	Attribut:	Intervalle d'essai par défaut d'interconnexion
8.4	(o)	Attribut:	Nombre de surveillances d'essai d'interconnexion
8.5	(o)	Attribut:	Intervalle d'interrogation de statut de liaison d'interconnexion
8.6	(o)	Attribut:	Nombre de répétitions d'interrogation de statut de liaison d'interconnexion
8.7	(m)	Attribut:	Etat de topologie d'interconnexion
9.	(c)	Contrainte:	Rôle d'interconnexion = CLIENT
9.1	(m)	Attribut:	Intervalle d'arrêt de la liaison d'interconnexion
9.2	(m)	Attribut:	Intervalle de fonctionnement de la liaison d'interconnexion
9.3	(m)	Attribut:	Nombre de modifications de la liaison d'interconnexion

SERVICES:

1	(m)	OpsService:	Démarrage MIM
2	(m)	OpsService:	Arrêt MIM
3	(o)	OpsService:	Modification de l'état d'interconnexion
4	(m)	OpsService:	Démarrage MIC
5	(m)	OpsService:	Arrêt MIC
5	(o)	OpsService:	Lecture MIM
6	(o)	OpsService:	Lecture MIC

6.3 Attributs

ID de domaine

Cet attribut clé définit le domaine de redondance représentant l'anneau auquel appartient l'objet de MRP. Il prend l'ID de domaine par défaut ou est spécifié en tant qu'ID unique par l'ingénierie.

Type d'attribut: UUID

Nom de domaine

Cet attribut définit le domaine de redondance représentant l'anneau auquel appartient l'objet de redondance du support. Il prend le nom de domaine par défaut ou est spécifié en tant qu'ID unique par l'ingénierie.

Type d'attribut: VisibleString[240]

ID 1 du port d'anneau

Cet attribut spécifie un port d'un commutateur affecté en tant que port en anneau 1 dans le domaine de redondance référencé par la valeur de l'attribut Domain ID.

Type d'attribut: Unsigned16

ID 2 du port d'anneau

Cet attribut spécifie un autre port d'un commutateur différent du port en anneau 1 affecté en tant que port en anneau 2 dans le domaine de redondance référencé par la valeur de l'attribut Domain ID.

Type d'attribut: Unsigned16

ID VLAN

L'objet de MRP peut utiliser cet attribut facultatif, qui spécifie son identifiant VLAN dans le domaine de redondance.

Type d'attribut: Unsigned16

Rôle attendu

Cet attribut spécifie le rôle de l'objet de MRP dans le domaine de redondance.

Type d'attribut: Unsigned16

Valeurs autorisées: MANAGER (Auto), MANAGER, CLIENT

Priorité du gestionnaire

Cet attribut doit contenir la priorité du MRM. Une valeur inférieure indique une priorité élevée, de 0x0000 (priorité la plus élevée) à 0xFFFF (priorité la plus basse) par incréments de 0x0001.

Type d'attribut: Unsigned16

Intervalle de modification de la topologie

Cet attribut spécifie l'intervalle d'envoi des trames MRP_TopologyChange.

Type d'attribut: Unsigned16

Nombre de répétitions de modification de la topologie

Cet attribut spécifie le nombre d'intervalles contrôlant les transmissions répétées des trames MRP_TopologyChange.

Type d'attribut: Unsigned16

Intervalle d'essai court

Cet attribut spécifie le court intervalle d'envoi des trames MRP_Test sur les ports en anneau à la suite des changements de liaison dans l'anneau.

Type d'attribut: Unsigned16

Intervalle d'essai par défaut

Cet attribut spécifie l'intervalle par défaut d'envoi de trames MRP_Test sur les ports en anneau.

Type d'attribut: Unsigned16

Nombre de surveillances d'essai

Cet attribut spécifie le nombre d'intervalles pour la surveillance de la réception des trames MRP_Test.

Type d'attribut: Unsigned16

MRC non bloquant pris en charge

Cet attribut spécifie la capacité du MRM à prendre en charge les MRC sans prise en charge de l'état du port BLOCKED dans l'anneau.

Type d'attribut: Boolean

Nombre étendu de surveillances d'essai

Cet attribut spécifie le nombre d'intervalles étendus pour la surveillance de la réception des trames MRP_Test.

Type d'attribut: Unsigned16

Modification de la liaison de réaction

Cet attribut facultatif indique si le MRM réagit immédiatement aux trames MRP_LinkChange.

Type d'attribut: Boolean

Vérification de la redondance du support

Cet attribut indique si la surveillance de l'état du MRM est activée (TRUE) ou désactivée (FALSE) dans le domaine de redondance.

Type d'attribut: Boolean

Etat réel du rôle

Cet attribut spécifie le rôle en cours de l'objet de MRP dans le domaine de redondance.

Type d'attribut: Unsigned16

Valeurs autorisées: MANAGER (Auto), MANAGER, CLIENT, UNDEFINED

Etat réel de l'anneau

Cet attribut spécifie l'état en cours de l'anneau de l'objet de MRP dans le domaine de redondance. L'une des valeurs suivantes doit être affectée à l'état de l'anneau:

OPEN: L'anneau est ouvert à cause d'une liaison ou d'une défaillance du MRC en son sein.

CLOSED: L'anneau est fermé (fonctionnement normal, pas d'erreur).

UNDEFINED: Doit être défini si l'attribut Real Role State contient la valeur CLIENT (c'est-à-dire que l'objet de MRP a été reconfiguré en fonction du rôle du client).

Type d'attribut: Unsigned16

Valeurs autorisées: OPEN, CLOSED, UNDEFINED

Etat du port 1 du port de l'anneau

Cet attribut facultatif spécifie l'état en cours du port en anneau 1. L'état du port en anneau 1 doit être spécifié conformément aux états du port en anneau définis en 5.2.

Type d'attribut: Unsigned16

Valeurs autorisées: DISABLED, BLOCKED, FORWARDING

Etat du port 2 du port de l'anneau

Cet attribut facultatif spécifie l'état en cours du port en anneau 2. L'état du port en anneau 2 doit être spécifié conformément aux états du port en anneau définis en 5.2.

Type d'attribut: Unsigned16

Valeurs autorisées: DISABLED, BLOCKED, FORWARDING

Intervalle d'arrêt de la liaison

Cet attribut spécifie l'intervalle d'envoi de trames MRP_LinkDown sur les ports en anneau.

Type d'attribut: Unsigned16

Intervalle de fonctionnement de la liaison

Cet attribut spécifie l'intervalle d'envoi de trames MRP_LinkUp sur les ports en anneau.

Type d'attribut: Unsigned16

Nombre de modifications de la liaison

Cet attribut spécifie le nombre de trames MRP_LinkChange contrôlant la transmission répétée des trames MRP_LinkChange.

Type d'attribut: Unsigned16

Etat BLOCKED pris en charge

Cet attribut indique si le MRC prend en charge l'état BLOCKED sur ses ports en anneau.

Type d'attribut: Boolean

ID d'interconnexion

Cet attribut clé définit le domaine d'interconnexion représentant la topologie d'interconnexion à laquelle appartient l'objet d'interconnexion de MRP. Il est spécifié en tant qu'ID unique par l'ingénierie.

Type d'attribut: Unsigned16

Nom d'interconnexion

Cet attribut spécifie le nom du domaine d'interconnexion référencé par la valeur de l'attribut Interconnection ID. Il prend le nom de l'interconnexion par défaut ou est spécifié en tant que nom de l'interconnexion par l'ingénierie.

Type d'attribut: VisibleString[240]

ID de port d'interconnexion

Cet attribut spécifie un port d'un commutateur affecté en tant que port d'interconnexion dans le domaine d'interconnexion référencé par la valeur de l'attribut Interconnection ID.

Type d'attribut: Unsigned16

Etat de port d'interconnexion

Cet attribut facultatif spécifie l'état en cours du port d'interconnexion. L'état du port d'interconnexion doit être spécifié conformément aux états du port d'interconnexion définis en 5.2.

Type d'attribut: Unsigned16

Valeurs autorisées: DISABLED, BLOCKED, FORWARDING

ID VLAN d'interconnexion

L'objet d'interconnexion de MRP peut utiliser cet attribut facultatif, qui spécifie son identifiant VLAN dans le domaine d'interconnexion.

Type d'attribut: Unsigned16

Mode d'interconnexion

Cet attribut spécifie le mode de l'objet d'interconnexion de MRP dans le domaine d'interconnexion.

LC_MODE: Mode de vérification de liaison. Le MIM analyse l'état d'interconnexion de l'anneau en utilisant les notifications de changement de liaison des MIC. Les trames MRP_InTest sont désactivées.

RC_MODE: Mode de vérification de l'anneau. Le MIM analyse l'état d'interconnexion de l'anneau en envoyant et en recevant des trames MRP_InTest. En outre, les notifications de changement de liaison des MIC peuvent être utilisées pour optimiser le temps de reprise.

Type d'attribut: Unsigned16

Valeurs autorisées: UNDEFINED, LC_MODE, RC_MODE

Rôle d'interconnexion

Cet attribut spécifie le rôle de l'objet d'interconnexion de MRP dans le domaine d'interconnexion.

Type d'attribut: Unsigned16

Valeurs autorisées: MANAGER, CLIENT

Intervalle de modification de topologie d'interconnexion

Cet attribut spécifie l'intervalle d'envoi des trames MRP_InTopologyChange.

Type d'attribut: Unsigned16

Nombre de répétitions de modification de topologie d'interconnexion

Cet attribut spécifie le nombre d'intervalles contrôlant les transmissions répétées des trames MRP_InTopologyChange.

Type d'attribut: Unsigned16

Intervalle d'essai par défaut d'interconnexion

Cet attribut spécifie l'intervalle par défaut d'envoi de trames MRP_InTest sur les ports en anneau et sur le port d'interconnexion.

Type d'attribut: Unsigned16

Nombre de surveillances d'essai d'interconnexion

Cet attribut spécifie le nombre d'intervalles pour la surveillance de la réception des trames MRP_InTest.

Type d'attribut: Unsigned16

Intervalle d'interrogation de statut de liaison d'interconnexion

Cet attribut spécifie l'intervalle d'envoi des trames MRP_InLinkStatusPoll.

Type d'attribut: Unsigned16

Nombre de répétitions d'interrogation de statut de liaison d'interconnexion

Cet attribut spécifie le nombre d'intervalles contrôlant les transmissions répétées des trames MRP_InLinkStatusPoll.

Type d'attribut: Unsigned16

Etat de topologie d'interconnexion

Cet attribut spécifie l'état en cours de topologie d'interconnexion de l'objet d'interconnexion de MRP dans le domaine d'interconnexion. L'une des valeurs suivantes doit être affectée à l'état de topologie d'interconnexion:

UNDEFINED: L'appareil ne dispose (pour l'instant) d'aucune information sur la topologie d'interconnexion;

OPEN: La topologie d'interconnexion est ouverte à cause d'une liaison ou d'une défaillance du MIC en son sein;

CLOSED: La topologie d'interconnexion est fermée (fonctionnement normal, pas d'erreur).

Type d'attribut: Unsigned16

Valeurs autorisées: UNDEFINED, OPEN, CLOSED

Intervalle d'arrêt de la liaison d'interconnexion

Cet attribut spécifie l'intervalle d'envoi de trames MRP_InLinkDown sur les ports en anneau.

Type d'attribut: Unsigned16

Intervalle de fonctionnement de la liaison d'interconnexion

Cet attribut spécifie l'intervalle d'envoi de trames MRP_InLinkUp sur les ports en anneau.

Type d'attribut: Unsigned16

Nombre de modifications de la liaison d'interconnexion

Cet attribut spécifie le nombre de trames MRP_InLinkChange contrôlant la transmission répétée des trames MRP_InLinkChange.

Type d'attribut: Unsigned16

7 Spécification de service du MRP

7.1 Démarrage MRM

Le service Start MRM permet de créer une instance locale de la machine de protocole du MRM.

Le Tableau 3 présente les paramètres du service.

Tableau 3 – Service Start MRM du MRP

Nom du paramètre	Req	Cnf
Argument	M	
ID de domaine	M	
ID 1 du port d'anneau	M	
ID 2 du port d'anneau	M	
ID VLAN	U	
Priorité du gestionnaire	U	
Intervalle de modification de la topologie	U	
Nombre de répétitions de modification de la topologie	U	
Intervalle d'essai court	U	
Intervalle d'essai par défaut	U	
Nombre de surveillances d'essai	U	
MRC non bloquant pris en charge	U	
Nombre étendu de surveillances d'essai	U	
Modification de la liaison de réaction	U	
Vérification de la redondance du support	U	
Résultat (+)		S
ID de domaine		M
Résultat (-)		S
ID de domaine		M
Code d'erreur		M
La signification de Req, Cnf, M, U et S, est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

ID 1 du port d'anneau

Ce paramètre contient l'ID du premier port en anneau.

ID 2 du port d'anneau

Ce paramètre contient l'ID du deuxième port en anneau.

ID VLAN

Ce paramètre facultatif contient la valeur de l'identifiant VLAN.

Priorité du gestionnaire

Ce paramètre contient la valeur de priorité du gestionnaire.

Intervalle de modification de la topologie

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_TopologyChange.

Nombre de répétitions de modification de la topologie

Ce paramètre contient la valeur du nombre d'intervalles contrôlant les transmissions répétées des trames MRP_TopologyChange.

Intervalle d'essai court

Ce paramètre contient la valeur du court intervalle d'envoi des trames MRP_Test sur les ports en anneau à la suite des changements de liaison dans l'anneau.

Intervalle d'essai par défaut

Ce paramètre contient la valeur de l'intervalle par défaut d'envoi de trames MRP_Test sur les ports en anneau.

Nombre de surveillances d'essai

Ce paramètre contient la valeur du nombre d'intervalles pour la surveillance de la réception des trames MRP_Test.

MRC non bloquant pris en charge

Ce paramètre spécifie la capacité du MRM à prendre en charge les MRC sans prise en charge de l'état du port BLOCKED dans l'anneau.

Nombre étendu de surveillances d'essai

Ce paramètre facultatif contient la valeur du nombre d'intervalles étendus pour la surveillance de la réception des trames MRP_Test.

Modification de la liaison de réaction

Ce paramètre facultatif spécifie si le MRM réagit aux trames MRP_LinkChange.

Vérification de la redondance du support

Ce paramètre permet d'activer ou de désactiver la surveillance de l'état du MRM.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: DOMAIN_ID_MISMATCH, ROLE_NOT_SUPPORTED, INVALID_RINGPORT

7.2 Arrêt MRM

Ce service doit être utilisé pour arrêter la machine de protocole du MRM. Les états du port en anneau et la fonction de commutation demeurent.

Le Tableau 4 présente les paramètres du service.

Tableau 4 – Service Stop MRM du MRP

Nom du paramètre	Req	Cnf
Argument	M	
ID de domaine	M	
Résultat (+)		S
ID de domaine		M
Résultat (-)		S
ID de domaine		M
Code d'erreur		M
La signification de Req, Cnf, M et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: DOMAIN_ID_MISMATCH

7.3 Modification de l'état

Ce service doit être utilisé pour indiquer un changement d'état du domaine du MRP.

Le Tableau 5 présente les paramètres du service.

Tableau 5 – Service Change State du MRP

Nom du paramètre	Ind
Argument	M
ID de domaine	M
Liste des types d'erreurs	M
La signification de M et Ind est spécifiée dans l'ISO/IEC 10164-1.	

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Liste des types d'erreurs

Cet attribut est composé des éléments suivants:

Type d'erreur

Cet attribut identifie une erreur de redondance du support.

Type d'attribut: Unsigned16

Valeurs autorisées: MANAGER_ROLE_FAIL, RING_OPEN, MULTIPLE_MANAGERS

Apparition

Cet attribut indique si l'erreur apparaît ou disparaît.

Type d'attribut: Boolean

Valeurs autorisées: TRUE, FALSE

7.4 Démarrage MRC

Le service Start MRC permet de créer une instance de la machine de protocole du MRC.

Le Tableau 6 présente les paramètres du service.

Tableau 6 – Service Start MRC du MRP

Nom du paramètre	Req	Cnf
Argument	M	
ID de domaine	M	
ID 1 du port d'anneau	M	
ID 2 du port d'anneau	M	
ID VLAN	U	
Intervalle d'arrêt de la liaison	U	
Intervalle de fonctionnement de la liaison	U	
Nombre de modifications de la liaison	U	
Etat BLOCKED pris en charge	U	
Résultat (+)		S
ID de domaine		M
Résultat (-)		S
ID de domaine		M
Code d'erreur		M
La signification de Req, Cnf, M, U et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

ID 1 du port d'anneau

Ce paramètre contient l'ID du premier port en anneau.

ID 2 du port d'anneau

Ce paramètre contient l'ID du deuxième port en anneau.

ID VLAN

Ce paramètre facultatif contient la valeur de l'identifiant VLAN.

Intervalle d'arrêt de la liaison

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_LinkDown sur les ports en anneau.

Intervalle de fonctionnement de la liaison

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_LinkUp sur les ports en anneau.

Nombre de modifications de la liaison

Ce paramètre contient la valeur du nombre de trames MRP_LinkChange contrôlant les transmissions répétées des trames MRP_LinkUp ou MRP_LinkDown.

Etat BLOCKED pris en charge

Ce paramètre indique si le MRC prend en charge l'état BLOCKED sur ses ports en anneau.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: DOMAIN_ID_MISMATCH, ROLE_NOT_SUPPORTED, INVALID_RINGPORT

7.5 Arrêt MRC

Ce service doit être utilisé pour arrêter la machine de protocole du MRC. Les états du port en anneau et la fonction de commutation demeurent. Le Tableau 7 présente les paramètres du service.

Tableau 7 – Service Arrêt MRC du MRP

Nom du paramètre	Req	Cnf
Argument	M	
ID de domaine	M	
Résultat (+)		S
ID de domaine		M
Résultat (-)		S
ID de domaine		M
Code d'erreur		M
La signification de Req, Cnf, M et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: DOMAIN_ID_MISMATCH

7.6 Lecture MRM

Le service facultatif Read MRM permet de lire l'état en cours de la machine de protocole du MRM.

Le Tableau 8 présente les paramètres du service.

Tableau 8 – Service Read MRM du MRP

Nom du paramètre	Req	Rsp
Argument	M	
ID de domaine	M	
Résultat (+)		S
ID de domaine		M
ID 1 du port d'anneau		M
ID 2 du port d'anneau		M
ID VLAN		U
Priorité du gestionnaire		M
Vérification de la redondance du support		M
Etat réel du rôle		M
Etat réel de l'anneau		M
Etat du port 1 du port de l'anneau		U
Etat du port 2 du port de l'anneau		U
Intervalle de modification de la topologie		U
Nombre de répétitions de modification de la topologie		U
Intervalle d'essai court		U
Intervalle d'essai par défaut		U
Nombre de surveillances d'essai		U
MRC non bloquant pris en charge		U
Nombre étendu de surveillances d'essai		U
Modification de la liaison de réaction		U
Résultat (-)		S
ID de domaine		M
Code d'erreur		M
La signification de Req, Rsp, M, U et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

ID 1 du port d'anneau

Ce paramètre contient l'ID du premier port en anneau.

ID 2 du port d'anneau

Ce paramètre contient l'ID du deuxième port en anneau.

ID VLAN

Ce paramètre facultatif contient la valeur de l'identifiant VLAN.

Priorité du gestionnaire

Ce paramètre contient la valeur de priorité du gestionnaire.

Vérification de la redondance du support

Ce paramètre indique si la surveillance du MRM est activée ou désactivée

Intervalle de modification de la topologie

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_TopologyChange.

Nombre de répétitions de modification de la topologie

Ce paramètre contient la valeur du nombre d'intervalles contrôlant la transmission répétée des trames MRP_TopologyChange.

Intervalle d'essai court

Ce paramètre contient la valeur du court intervalle d'envoi des trames MRP_Test sur les ports en anneau à la suite des changements de liaison dans l'anneau.

Intervalle d'essai par défaut

Ce paramètre contient la valeur de l'intervalle par défaut d'envoi de trames MRP_Test sur les ports en anneau.

Nombre de surveillances d'essai

Ce paramètre contient la valeur du nombre d'intervalles pour la surveillance de la réception des trames MRP_Test.

MRC non bloquant pris en charge

Ce paramètre indique la capacité du MRM à prendre en charge les MRC sans prise en charge de l'état du port BLOCKED dans l'anneau.

Nombre étendu de surveillances d'essai

Ce paramètre facultatif contient la valeur du nombre d'intervalles étendus pour la surveillance de la réception des trames MRP_Test.

Etat réel du rôle

Cet attribut contient le rôle en cours de l'objet de MRP dans le domaine de redondance.

Etat réel de l'anneau

Cet attribut contient l'état en cours de l'anneau de l'objet de MRP dans le domaine de redondance.

Etat du port 1 du port de l'anneau

Cet attribut facultatif contient l'état en cours du port en anneau 1.

Etat du port 2 du port de l'anneau

Cet attribut facultatif contient l'état en cours du port en anneau 2.

Modification de la liaison de réaction

Ce paramètre facultatif indique si le MRM réagit aux trames MRP_LinkChange.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: DOMAIN_ID_MISMATCH, MANAGER_READ_FAIL

7.7 Lecture MRC

Le service facultatif Read MRC permet de lire l'état en cours de la machine de protocole du MRC.

Le Tableau 9 présente les paramètres du service.

Tableau 9 – Service Read MRC du MRP

Nom du paramètre	Req	Rsp
Argument	M	
ID de domaine	M	
Résultat (+)		S
ID de domaine		M
ID 1 du port d'anneau		M
ID 2 du port d'anneau		M
ID VLAN		U
Etat du port 1 du port de l'anneau		U
Etat du port 2 du port de l'anneau		U
Intervalle d'arrêt de la liaison		U
Intervalle de fonctionnement de la liaison		U
Nombre de modifications de la liaison		U
Etat BLOCKED pris en charge		U
Résultat (-)		S
ID de domaine		M
Code d'erreur		M
La signification de Req, Rsp, M, U et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

ID 1 du port d'anneau

Ce paramètre contient l'ID du premier port en anneau.

ID 2 du port d'anneau

Ce paramètre contient l'ID du deuxième port en anneau.

ID VLAN

Ce paramètre facultatif contient la valeur de l'identifiant VLAN.

Intervalle d'arrêt de la liaison

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_LinkDown sur les ports en anneau.

Intervalle de fonctionnement de la liaison

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_LinkUp sur les ports en anneau.

Nombre de modifications de la liaison

Ce paramètre contient la valeur du nombre de trames MRP_LinkChange contrôlant la transmission répétée des trames MRP_LinkUp ou MRP_LinkDown.

Etat du port 1 du port de l'anneau

Cet attribut facultatif contient l'état en cours du port en anneau 1.

Etat du port 2 du port de l'anneau

Cet attribut facultatif contient l'état en cours du port en anneau 2.

Etat BLOCKED pris en charge

Ce paramètre indique si le MRC prend en charge l'état BLOCKED sur ses ports en anneau.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID de domaine

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: DOMAIN_ID_MISMATCH, CLIENT_READ_FAIL

7.8 Démarrage MIM

Le service Start MIM permet de créer une instance locale de la machine de protocole du MIM.

Le Tableau 10 présente les paramètres du service.

Tableau 10 – Service Start MIM du MRP

Nom du paramètre	Req	Cnf
Argument	M	
ID d'interconnexion	M	
Nom d'interconnexion	M	
ID de port d'interconnexion	M	
ID VLAN d'interconnexion	U	
Mode d'interconnexion	M	
Intervalle de modification de topologie d'interconnexion	U	
Nombre de répétitions de modification de topologie d'interconnexion	U	
Intervalle d'essai par défaut d'interconnexion	U	
Nombre de surveillances d'essai d'interconnexion	U	
Intervalle d'interrogation de statut de liaison d'interconnexion	U	
Nombre de répétitions d'interrogation de statut de liaison d'interconnexion	U	
Résultat (+)		S
ID d'interconnexion		M
Résultat (-)		S
ID d'interconnexion		M
Code d'erreur		M
La signification de Req, Cnf, M, U et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Nom d'interconnexion

Ce paramètre contient le nom du domaine d'interconnexion.

ID de port d'interconnexion

Ce paramètre contient l'ID du port d'interconnexion.

ID VLAN d'interconnexion

Ce paramètre facultatif contient la valeur de l'identifiant VLAN.

Mode d'interconnexion

Ce paramètre contient le mode de fonctionnement exigé du MIM.

Intervalle de modification de topologie d'interconnexion

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_InTopologyChange.

Nombre de répétitions de modification de topologie d'interconnexion

Ce paramètre contient la valeur du nombre d'intervalles contrôlant les transmissions répétées des trames MRP_InTopologyChange.

Intervalle d'essai par défaut d'interconnexion

Ce paramètre contient la valeur de l'intervalle par défaut d'envoi de trames MRP_InTest sur les ports en anneau et sur le port d'interconnexion.

Nombre de surveillances d'essai d'interconnexion

Ce paramètre contient la valeur du nombre d'intervalles pour la surveillance de la réception des trames MRP_InTest.

Intervalle d'interrogation de statut de liaison d'interconnexion

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_InLinkStatusPoll.

Nombre de répétitions du statut de liaison d'interconnexion

Ce paramètre contient la valeur du nombre d'intervalles contrôlant les transmissions répétées des trames MRP_InLinkStatusPoll.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: INTERCONNECTION_ID_MISMATCH, ROLE_NOT_SUPPORTED, INVALID_INTERCONNECTIONPORT

7.9 Arrêt MIM

Ce service doit être utilisé pour arrêter la machine de protocole du MIM. L'état du port d'interconnexion et la fonction de commutation demeurent.

Le Tableau 11 présente les paramètres du service.

Tableau 11 – Service Stop MIM du MRP

Nom du paramètre	Req	Cnf
Argument	M	
ID d'interconnexion	M	
Résultat (+)		S
ID d'interconnexion		M
Résultat (-)		S
ID d'interconnexion		M
Code d'erreur		M
La signification de Req, Cnf, M et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: INTERCONNECTION_ID_MISMATCH

7.10 Modification de l'état d'interconnexion

Ce service facultatif doit être utilisé pour indiquer un changement d'état du domaine d'interconnexion du MRP.

Le Tableau 12 présente les paramètres du service.

Tableau 12 – Service Interconnection Change State du MRP

Nom du paramètre	Ind
Argument	M
ID d'interconnexion	M
Liste des types d'erreurs	M
La signification de M et Ind est spécifiée dans l'ISO/IEC 10164-1.	

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Liste des types d'erreurs

Cet attribut est composé des éléments suivants:

Type d'erreur

Cet attribut identifie une erreur d'interconnexion de redondance du support.

Type d'attribut: Unsigned16

Valeurs autorisées: INTERCONNECTION_OPEN

Apparition

Cet attribut indique si l'erreur apparaît ou disparaît.

Type d'attribut: Boolean

Valeurs autorisées: TRUE, FALSE

7.11 Démarrage MIC

Le service Start MIC permet de créer une instance de la machine de protocole du MIC.

Le Tableau 13 présente les paramètres du service.

Tableau 13 – Service Start MIC du MRP

Nom du paramètre	Req	Cnf
Argument	M	
ID d'interconnexion	M	
Nom d'interconnexion	M	
ID de port d'interconnexion	M	
ID VLAN d'interconnexion	U	
Mode d'interconnexion	M	
Intervalle d'arrêt de la liaison d'interconnexion	U	
Intervalle de fonctionnement de la liaison d'interconnexion	U	
Nombre de modifications de la liaison d'interconnexion	U	
Résultat (+)		S
ID d'interconnexion		M
Résultat (-)		S
ID d'interconnexion		M
Code d'erreur		M
La signification de Req, Cnf, M, U et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Nom d'interconnexion

Ce paramètre contient le nom du domaine d'interconnexion.

ID de port d'interconnexion

Ce paramètre contient l'ID du port d'interconnexion.

ID VLAN d'interconnexion

Ce paramètre facultatif contient la valeur de l'identifiant VLAN.

Mode d'interconnexion

Ce paramètre contient le mode de fonctionnement exigé du MIC.

Intervalle d'arrêt de la liaison d'interconnexion

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_InLinkDown sur les ports en anneau.

Intervalle de fonctionnement de la liaison d'interconnexion

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_InLinkUp sur les ports en anneau.

Nombre de modifications de la liaison d'interconnexion

Ce paramètre contient la valeur du nombre de trames MRP_InLinkChange contrôlant les transmissions répétées des trames MRP_InLinkUp ou MRP_InLinkDown.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: INTERCONNECTION_ID_MISMATCH, ROLE_NOT_SUPPORTED, INVALID_INTERCONNECTIONPORT

7.12 Arrêt MIC

Ce service doit être utilisé pour arrêter la machine de protocole du MIC. L'état du port d'interconnexion et la fonction de commutation demeurent.

Le Tableau 14 présente les paramètres du service.

Tableau 14 – Service Stop MIC du MRP

Nom du paramètre	Req	Cnf
Argument	M	
ID d'interconnexion	M	
Résultat (+)		S
ID d'interconnexion		M
Résultat (-)		S
ID d'interconnexion		M
Code d'erreur		M
La signification de Req, Cnf, M et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: INTERCONNECTION_ID_MISMATCH

7.13 Lecture MIM

Le service facultatif Read MIM permet de lire l'état en cours de la machine de protocole du MIM.

Le Tableau 15 présente les paramètres du service.

Tableau 15 – Service Read MIM du MRP

Nom du paramètre	Req	Rsp
Argument	M	
ID d'interconnexion	M	
Résultat (+)		S
ID d'interconnexion		M
Nom d'interconnexion		M
ID de port d'interconnexion		M
Etat de port d'interconnexion		U
ID VLAN d'interconnexion		U
Mode d'interconnexion		M
Intervalle de modification de topologie d'interconnexion		U
Nombre de répétitions de modification de topologie d'interconnexion		U
Intervalle d'essai par défaut d'interconnexion		U
Nombre de surveillances d'essai d'interconnexion		U
Intervalle d'interrogation de statut de liaison d'interconnexion		U
Nombre de répétitions d'interrogation de statut de liaison d'interconnexion		U
Etat de topologie d'interconnexion		M
Résultat (-)		S
ID d'interconnexion		M
Code d'erreur		M
La signification de Req, Rsp, M, U et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Nom d'interconnexion

Ce paramètre contient le nom du domaine d'interconnexion.

ID de port d'interconnexion

Ce paramètre contient l'ID du port d'interconnexion.

Etat de port d'interconnexion

Cet attribut facultatif contient l'état en cours du port d'interconnexion.

ID VLAN d'interconnexion

Ce paramètre facultatif contient la valeur de l'identifiant VLAN.

Mode d'interconnexion

Ce paramètre contient le mode de fonctionnement du MIM.

Intervalle de modification de topologie d'interconnexion

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_InTopologyChange.

Nombre de répétitions de modification de topologie d'interconnexion

Ce paramètre contient la valeur du nombre d'intervalles contrôlant la transmission répétée des trames MRP_InTopologyChange.

Intervalle d'essai par défaut d'interconnexion

Ce paramètre contient la valeur de l'intervalle par défaut d'envoi de trames MRP_InTest sur les ports en anneau.

Nombre de surveillances d'essai d'interconnexion

Ce paramètre contient la valeur du nombre d'intervalles pour la surveillance de la réception des trames MRP_InTest.

Intervalle d'interrogation de statut de liaison d'interconnexion

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_InLinkStatusPoll.

Nombre de répétitions d'interrogation de statut de liaison d'interconnexion

Ce paramètre contient la valeur du nombre d'intervalles pour les transmissions répétées des trames MRP_InLinkStatusPoll.

Etat de topologie d'interconnexion

Cet attribut contient l'état en cours de topologie d'interconnexion de l'objet d'interconnexion de MRP dans le domaine d'interconnexion.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: INTERCONNECTION_ID_MISMATCH, MANAGER_READ_FAIL

7.14 Lecture MIC

Le service facultatif Read MIC permet de lire l'état en cours de la machine de protocole du MIC.

Le Tableau 16 présente les paramètres du service.

Tableau 16 – Service Read MIC du MRP

Nom du paramètre	Req	Rsp
Argument	M	
ID d'interconnexion	M	
Résultat (+)		S
ID d'interconnexion		M
Nom d'interconnexion		M
ID de port d'interconnexion		M
Etat de port d'interconnexion		U
ID VLAN d'interconnexion		U
Mode d'interconnexion		M
Intervalle d'arrêt de la liaison d'interconnexion		U
Intervalle de fonctionnement de la liaison d'interconnexion		U
Nombre de modifications de la liaison d'interconnexion		U
Résultat (-)		S
ID d'interconnexion		M
Code d'erreur		M
La signification de Req, Rsp, M, U et S est spécifiée dans l'ISO/IEC 10164-1.		

Argument

L'argument doit acheminer les paramètres spécifiques au service de la demande de service.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Résultat (+)

Ce paramètre indique que la demande de service a abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Nom d'interconnexion

Ce paramètre contient le nom du domaine d'interconnexion.

ID de port d'interconnexion

Ce paramètre contient l'ID du port d'interconnexion.

Etat de port d'interconnexion

Cet attribut facultatif contient l'état en cours du port d'interconnexion.

ID VLAN d'interconnexion

Ce paramètre facultatif contient la valeur de l'identifiant VLAN.

Mode d'interconnexion

Ce paramètre contient le mode de fonctionnement du MIC.

Intervalle d'arrêt de la liaison d'interconnexion

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_InLinkDown sur les ports en anneau.

Intervalle de fonctionnement de la liaison d'interconnexion

Ce paramètre contient la valeur de l'intervalle d'envoi de trames MRP_InLinkUp sur les ports en anneau.

Nombre de modifications de la liaison d'interconnexion

Ce paramètre contient la valeur du nombre de trames MRP_InLinkChange contrôlant la transmission répétée des trames MRP_InLinkUp ou MRP_InLinkDown.

Résultat (-)

Ce paramètre indique que la demande de service n'a pas abouti.

ID d'interconnexion

Il s'agit de l'attribut clé qui permet d'identifier l'instance de la machine de protocole d'interconnexion.

Code d'erreur

Le paramètre Résultat contient le code d'erreur spécifique.

Type: Unsigned16

Valeurs autorisées: INTERCONNECTION_ID_MISMATCH, CLIENT_READ_FAIL

8 Spécification du protocole MRP

8.1 Description du PDU

8.1.1 Types de données de base

Les conventions de la présente spécification sont conformes à l'IEC 61158-6-10:2014, 4.2. La notation et le codage des types de données de base sont conformes à l'IEC 61158-6-10:2014, 4.1.2 et à l'IEC 61158-6-10:2014, 4.2.

8.1.2 Référence de syntaxe abstraite du DLPDU

La syntaxe de transfert et le codage de la spécification du protocole MRP sont conformes à l'IEC 61158-6-10:2014, 4.2.

Le codage et le décodage des champs du Tableau 17 doivent être conformes à l'ISO/IEC/IEEE 8802-3 (IEEE 802.3) pour le DLPDU.

Tableau 17 – Syntaxe du DLPDU du MRP pour l'ISO/IEC/IEEE 8802-3 (IEEE 802.3)

Nom du DLPDU	Structure du DLPDU
DLPDU	Preamble ^a , StartFrameDelimiter, DestinationAddress, SourceAddress, DLSDU ^b , DLPDU_Padding ^c , FrameCheckSequence
DLSDU	[VLAN] ^d , LT, MRP-PDU
VLAN	LT(=0x8100), TagControlInformation
NOTE 1 Conformément à l'ISO/IEC/IEEE 8802-3 (IEEE 802.3), la longueur minimale des DLPDU est de 64 octets (excepté pour Preamble et StartFrameDelimiter). NOTE 2 La longueur minimale des trames ISO/IEC/IEEE 8802-3 (IEEE 802.3) dotées de la balise VLAN est désormais de 68 octets, afin d'assurer une longueur minimale de trame de 64 octets après la suppression de la balise VLAN par un pont.	
^a Le champ contient au moins 7 octets. ^b La taille DLSDU minimale est de 2 octets. ^c Le nombre d'octets de remplissage doit être compris entre 0 et 46, selon la longueur du DLSDU. La valeur doit être nulle. ^d Le champ VLAN peut être ignoré en cas de transport optimisé. Le champ VLAN peut être défini par le codeur, mais peut être rejeté par les ponts intermédiaires. Le décodeur doit accepter les DLPDU avec ou sans champ VLAN.	

8.1.3 Codage du champ de DLPDU SourceAddress

Ce champ doit être codé en tant que type de données octetString[6]. La valeur du champ SourceAddress doit être conforme à l'adresse MAC de l'IEEE 802; voir IEEE 802-1D:2004, Article 7. L'adresse MAC du port est utilisée pour le DLPDU du MRP. L'adresse MAC de l'interface doit être différente de l'adresse MAC du port.

8.1.4 Codage du champ de DLPDU DestinationAddress

Ce champ doit être codé en tant que type de données octetString[6].

L'identificateur unique d'organisation de l'IEEE correspondant au MRP est 00-15-4E. Il doit être configuré conformément au Tableau 18.

Tableau 18 – OUI du MRP

Valeur de l'OUI (hexadécimale)	Signification
00-15-4E	Monodiffusion individuelle administrée au niveau global
01-15-4E	Adresse (multidiffusion) de groupe administrée au niveau global
02-15-4E	Monodiffusion individuelle administrée au niveau local
03-15-4E	Adresse (multidiffusion) de groupe administrée au niveau local

Pour les MRP-PDU, la valeur de l'adresse de destination doit être définie conformément au Tableau 19.

Tableau 19 – MulticastMACAddress du MRP

Valeur de l'OUI (multidiffusion) (hexadécimale)	Valeur du code ExtensionIdentifieur (hexadécimale)	Signification
01-15-4E	00-00-00	Réservé
01-15-4E	00-00-01	MC_TEST, utilisée pour les trames MRP_Test
01-15-4E	00-00-02	MC_CONTROL, utilisée pour les trames MRP_LinkChange, MRP_TopologyChange et MRP_Option
01-15-4E	00-00-03	MC_INTEST, utilisée pour les trames MRP_InTest
01-15-4E	00-00-04	MC_INCONTROL, utilisée pour les trames MRP_InLinkChange, MRP_InTopologyChange et MRP_InLinkStatusPoll
01-15-4E	00-00-05 à FF-FF-FF	Réservé

NOTE 1 L'octet 1 contient le bit d'adresse individuelle/de groupe (LSB).

NOTE 2 L'adresse multidiffusion de destination MC_TEST est également utilisée pour les trames MRP_TestMgrNack et MRP_TestPropagate.

8.1.5 Codage du champ TagControlInformation

Ce champ doit être codé conformément à l'IEEE 802.1Q en tant que type de données Unsigned16. Les bits individuels doivent avoir la signification suivante:

Bit 0 – 11: TagControlInformation.VLAN_Identifier

Ces bits doivent être codés conformément à l'IEEE 802.1Q.

NOTE VLAN_Identifier 0 signifie qu'aucun VLAN n'est utilisé.

Bit 12: TagControlInformation.CanonicalFormatIdentifier

Ces bits doivent être codés conformément à l'IEEE 802.1Q.

NOTE Le code CFI est une constante 0.

Bit 13 – 15: TagControlInformation.Priority

Ces bits doivent être codés conformément à l'IEEE 802.1Q.

Pour le MRP, l'utilisation du champ VLAN est facultative. En sa présence, la valeur de TagControlInformation.Priority doit être configurée conformément au Tableau 20.

Tableau 20 – Champ TagControlInformation.Priority du MRP

Valeur (hexadécimale)	Signification
0x07	MRP-PDU

8.1.6 Codage du champ LT

Ce champ doit être codé en tant que type de données Unsigned16 avec des valeurs conformes à l'ISO/IEC/IEEE 8802-3 (IEEE 802.3). Pour le MRP, la valeur doit être configurée conformément au Tableau 21.

Tableau 21 – Champ LT du MRP

Valeur (hexadécimale)	Signification
0x88E3	MRP-PDU

8.1.7 Syntaxe abstraite des APDU du MRP

Le Tableau 22 définit la syntaxe abstraite des MRP-PDU appelés APDU. L'ordre défini des octets doit être utilisé pour acheminer les APDU.

Tableau 22 – Syntaxe des APDU du MRP

Nom de l'APDU	Structure de l'APDU
MRP-PDU	MRP_Version, MRP_Type, MRP_Common, [MRP_Option], MRP_End, [Remplissage*] ^a
^a Si la trame contient moins de 64 octets, elle doit être étendue à 64 octets par remplissage, conformément à l'ISO/IEC/IEEE 8802-3 (IEEE 802.3).	

Le Tableau 23 définit les structures de substitution des éléments de la structure de l'APDU présentés au Tableau 22.

Tableau 23 – Substitutions au sein du MRP

Nom de substitution	Structure
MRP_Type	MRP_Test ^ MRP_LinkChange ^ MRP_TopologyChange ^ MRP_Option ^ MRP_InTest ^ MRP_InLinkChange ^ MRP_InLinkStatusPoll ^ MRP_InTopologyChange
MRP_Common	MRP_TLVHeader, MRP_SequenceID, MRP_DomainUUID
MRP_Option ^c	MRP_TLVHeader, MRP_OUI, MRP_SubOption1, [MRP_SubOption2*] , [Remplissage*] ^a
MRP_OUI	MRP_ManufacturerOUI ^ MRP_IECOUI
MRP_SubOption1	MRP_Ed1Type, MRP_Ed1ManufacturerData
MRP_SubOption2 ^d	MRP_TestMgrNAck ^ MRP_TestPropagate ^ MRP_AutoMgr ^ MRP_ManufacturerFkt
MRP_TestMgrNAck ^e	MRP_SubTLVHeader, MRP_Prio, MRP_SA, MRP_OtherMRMPrio, MRP_OtherMRMSA, [Remplissage*] ^a
MRP_TestPropagate ^e	MRP_SubTLVHeader, MRP_Prio, MRP_SA, MRP_OtherMRMPrio, MRP_OtherMRMSA, [Remplissage*] ^a
MRP_AutoMgr ^f	MRP_SubTLVHeader (=0x0300)
MRP_ManufacturerFkt	MRP_SubTLVHeader, MRP_ManufacturerData
MRP_End	MRP_TLVHeader (=0x0000)
MRP_Test ^b	MRP_TLVHeader, MRP_Prio, MRP_SA, MRP_PortRole, MRP_RingState, MRP_Transition, MRP_TimeStamp, [Remplissage*] ^a
MRP_TopologyChange	MRP_TLVHeader, MRP_Prio, MRP_SA, MRP_Interval, [Remplissage*] ^a
MRP_LinkChange	MRP_LinkUp ^ MRP_LinkDown
MRP_LinkDown	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_Interval, MRP_Blocked, [Remplissage*] ^a
MRP_LinkUp	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_Interval, MRP_Blocked, [Remplissage*] ^a
MRP_InTest	MRP_TLVHeader, MRP_InID, MRP_SA, MRP_PortRole, MRP_InState, MRP_Transition, MRP_TimeStamp, [Remplissage*] ^a
MRP_InLinkChange	MRP_InLinkUp ^ MRP_InLinkDown
MRP_InLinkDown	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_InID, MRP_Interval, [Remplissage*] ^a
MRP_InLinkUp	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_InID, MRP_Interval, [Remplissage*] ^a
MRP_InLinkStatusPoll	MRP_TLVHeader, MRP_SA, MRP_PortRole, MRP_InID, [Remplissage*] ^a
MRP_InTopologyChange	MRP_TLVHeader, MRP_SA, MRP_InID, MRP_Interval, [Remplissage*] ^a
^a L'alignement 32 bits doit être assuré. ^b Pour une connexion avec des systèmes d'automation à haute vitesse, la longueur du MRP_PDU de type MRP_Test est essentielle. Il convient de la maintenir à une valeur minimale. Consulter les normes relatives au système d'automation applicables pour les limitations de longueur de trame. ^c Les trames de MRP_Type = MRP_Option ne doivent pas contenir plus d'un MRP_SubOption2 avec un MRP_SubTLVHeader.Type entre 0x01 et 0xF0. ^d MRP_SubTLVHeader.Type dans l'ordre croissant. ^e A utiliser seulement dans MRP_Type = MRP_Option. ^f A utiliser seulement dans MRP_Type = MRP_Test. En outre, MRP_SubOption2 ne peut être spécifique à un fabricant qu'avec un MRP_SubTLVHeader.Type entre 0xF1 et 0xFF.	

8.1.8 Codage du champ MRP_TLVHeader

Le codage de ce champ doit être conforme à l'IEC 61158-6-10:2014, 4.2.1. La signification des bits individuels doit être la suivante:

Bit 0 – 7: MRP_TLVHeader.Length

La valeur contient le nombre d'octets consécutifs du bloc correspondant.

Bit 8 – 15: MRP_TLVHeader.Type

Ce champ doit être codé avec les valeurs conformes au Tableau 24.

Tableau 24 – MRP_TLVHeader.Type

Valeur (hexadécimale)	Signification	Utilisation
0x00	MRP_End (MRP_TLVHeader.Length doit être nul)	Obligatoire
0x01	MRP_Common	Obligatoire
0x02	MRP_Test	Obligatoire
0x03	MRP_TopologyChange	Obligatoire
0x04	MRP_LinkDown	Obligatoire
0x05	MRP_LinkUp	Obligatoire
0x06	MRP_InTest	Obligatoire en RC-mode
0x07	MRP_InTopologyChange	Obligatoire
0x08	MRP_InLinkDown	Obligatoire
0x09	MRP_InLinkUp	Obligatoire
0x0A	MRP_InLinkStatusPoll	Obligatoire en LC-mode
0x0B – 0x7E	Réservé	—
0x7F	MRP_Option (spécifique à l'organisation)	Facultative
0x80-0xFF	Réservé	—

8.1.9 Codage du champ MRP_SubTLVHeader

Le codage de ce champ doit être conforme à l'IEC 61158-6-10:2014, 4.2.1. La signification des bits individuels doit être la suivante:

Bit 0 – 7: MRP_SubTLVHeader.Length

La valeur contient le nombre d'octets consécutifs du bloc correspondant.

Bit 8 – 15: MRP_SubTLVHeader.Type

Ce champ doit être codé avec les valeurs conformes au Tableau 25.

Tableau 25 – MRP_SubTLVHeader.Type

Valeur (hexadécimale)	OUI	Signification	Utilisation
0x00	indifférent	Réservé	Obligatoire
0x01	ManufacturerOUI ou IEC OUI	MRP_TestMgrNAck	Obligatoire
0x02	ManufacturerOUI ou IEC OUI	MRP_TestPropagate	Obligatoire
0x03	ManufacturerOUI ou IEC OUI	MRP_AutoMgr	Obligatoire
0x04 – 0xF0	ManufacturerOUI ou IEC OUI	Réservée aux fonctions spécifiques de l'IEC	Facultative
0xF1 – 0xFF	ManufacturerOUI	Fonctions spécifiques à un fabricant	Facultative

8.1.10 Codage des champs MRP_Ed1Type et MRP_Ed1ManufacturerData

Le champ MRP_Ed1Type doit être codé en type de données Unsigned8.

Pour MRP_OUI = MRP_ManufacturerOUI, le contenu des champs doit être codé avec les valeurs conformes au Tableau 26.

Tableau 26 – MRP_Ed1Type et MRP_Ed1ManufacturerData

MRP_Ed1Type Valeur (hexadécimale)	MRP_Ed1ManufacturerData Sommaire	MRP_Ed1ManufacturerData Longueur (octets)	Utilisation
0x00	indifférent	2	Obligatoire
0x01 – 0x03	-	0	Obligatoire
0x04	indifférent	26	Obligatoire
0x05 – 0xFE	-	0	Obligatoire
0xFF	-	0	Obligatoire

Pour MRP_OUI = IECUI, le contenu des champs doit être codé avec les valeurs conformes au Tableau 27.

Tableau 27 – MRP_Ed1Type et MRP_Ed1ManufacturerData

MRP_Ed1Type Valeur (hexadécimale)	MRP_Ed1ManufacturerData Sommaire	MRP_Ed1ManufacturerData Longueur (octets)	Utilisation
0x00 – 0xFE	Réservé	0	Obligatoire
0xFF	indifférent	0	Obligatoire

8.1.11 Codage du champ MRP_Version

Ce champ doit être codé comme type de données Unsigned16 avec les valeurs conformes au Tableau 28.

Tableau 28 – MRP_Version

Valeur (décimale)	Signification
0	Réservé
1	Version initiale de MRP
2 ... 65 535	Réservé

8.1.12 Codage du champ MRP_SequenceID

Ce champ doit être codé en tant que type de données Unsigned16. Il est utilisé pour identifier la duplication de trames du MRP dans l'anneau. La plage de valeurs va de 0 à 65 535. Le processus d'application demandeur doit fournir un numéro de séquence unique à chaque demande de services en cours.

8.1.13 Codage du champ MRP_SA

Ce champ doit être codé en tant que type de données octetString[6]. La valeur du champ MRP_SA doit être conforme à l'adresse MAC de l'IEEE 802 et doit contenir l'adresse MAC de l'hôte de commutation émetteur (adresse MAC de l'interface).

8.1.14 Codage du champ MRP_OtherMRMSA

Ce champ doit être codé en tant que type de données octetString[6]. La valeur du champ MRP_OtherMRMSA doit être conforme à l'adresse MAC de l'IEEE 802 et doit être configurée conformément au Tableau 29.

Tableau 29 – Codage du champ MRP_OtherMRMSA

MRP_SubOption2	Valeur de MRP_OtherMRMSA
MRP_TestMgrNack	Adresse MAC de l'interface du gestionnaire à la priorité la plus basse dont la trame d'essai a été préalablement reçue.
MRP_TestPropagate	Adresse MAC de l'interface du gestionnaire à la priorité la plus élevée dont la trame MRP_TestMgrNack a été préalablement reçue.

8.1.15 Codage du champ MRP_Prio

Ce champ doit être codé en tant que type de données Unsigned16 et configuré conformément au Tableau 30.

Tableau 30 – MRP_Prio

Valeur (hexadécimale)	Signification
0x0000	Priorité la plus élevée du MRM
0x1000 – 0x7000	Priorités élevées du MRM
0x8000	Priorité par défaut du MRM
0x9000 – 0x9FFF	Priorités élevées du MRA
0xA000	Priorité par défaut du MRA
0xA001-0xF000	Priorités basses du MRA
0xFFFF	Priorité la plus basse du MRA
Autre	Réservé

NOTE Le fait d'avoir un ou plusieurs appareils configurés pour le rôle MRA et une MRM configurée manuellement dans le même anneau n'est pas pris en charge dans la présente édition de l'IEC 62439-2.

8.1.16 Codage du champ MRP_OtherMRMPrio

Ce champ doit être codé en tant que type de données Unsigned16. La valeur du champ MRP_OtherMRMPrio doit être configurée conformément au Tableau 31.

Tableau 31 – Codage du champ MRP_OtherMRMPrio

MRP_SubOption2	Valeur de MRP_OtherMRMPrio
MRP_TestMgrNack	Le champ est laissé vide et ne sera pas analysé par le destinataire.
MRP_TestPropagate	Priorité du gestionnaire dont la priorité est la plus élevée, dont la trame TestMgrNack a été préalablement reçue

8.1.17 Codage du champ MRP_PortRole

Ce champ doit être codé en tant que type de données Unsigned16. Le codage doit être conforme au Tableau 32.

Tableau 32 – MRP_PortRole

Valeur (hexadécimale)	Signification	Utilisation
0x0000	Port en anneau principal	La trame est envoyée sur le port en anneau principal
0x0001	Port en anneau secondaire	La trame est envoyée sur le port en anneau secondaire
0x0002	Port d'interconnexion	La trame est envoyée sur le port d'interconnexion
0x0003 – 0xFFFF	Réservé	

8.1.18 Codage du champ MRP_RingState

Ce champ doit être codé comme type de données Unsigned16 avec les valeurs conformes au Tableau 33.

Tableau 33 – MRP_RingState

Valeur (hexadécimale)	Signification	Utilisation
0x0000	Anneau ouvert	Le MRM est à l'état d'anneau ouvert
0x0001	Anneau fermé	Le MRM est à l'état d'anneau fermé
0x0002 – 0xFFFF	Réservé	—

8.1.19 Codage du champ MRP_Interval

Ce champ doit être codé comme type de données Unsigned16 avec les valeurs conformes au Tableau 34.

Tableau 34 – MRP_Interval

Valeur (hexadécimale)	Signification	Utilisation
0x0000 – 0x07D0	Intervalle de l'événement de changement de topologie suivant (en ms)	Obligatoire
0x07D1 – 0xFFFF	Intervalle de l'événement de changement de topologie suivant (en ms)	Facultative

8.1.20 Codage du champ MRP_Transition

Ce champ doit être codé comme type de données Unsigned16 avec les valeurs conformes au Tableau 35.

Tableau 35 – MRP_Transition

Valeur (hexadécimale)	Signification	Utilisation
0x0000 – 0xFFFF	Nombre de transitions entre les états d'anneau ouvert et fermé ou nombre de transitions entre l'état ouvert de topologie d'interconnexion et l'état fermé de topologie d'interconnexion .	Utilisée pour surveiller cette valeur grâce à un renifleur de paquet

8.1.21 Codage du champ MRP_TimeStamp

Ce champ doit être codé comme type de données Unsigned32 avec les valeurs conformes au Tableau 36.

Tableau 36 – MRP_TimeStamp

Valeur (hexadécimale)	Signification	Utilisation
0x00000000 – 0xFFFFFFFF	Valeur en cours du compteur local de 1 ms	La valeur est utilisée par le MRM pour déterminer le temps de déplacement maximal des trames MRP_Test dans un anneau. Le MIM utilise également cette valeur pour déterminer le temps de déplacement maximal des trames MRP_InTest dans une topologie d'interconnexion.

8.1.22 Codage du champ MRP_Blocked

Ce champ doit être codé comme type de données Unsigned16 avec les valeurs conformes au Tableau 37.

Tableau 37 – MRP_Blocked

Valeur (décimale)	Signification	Utilisation
0	Le MRC ne peut ni recevoir ni transférer les trames MRP_Test, MRP_LinkChange et MRP_TopologyChange au niveau d'un port en anneau dont l'état est BLOCKED	Facultative
1	Le MRC peut recevoir et transférer les trames MRP_Test, MRP_LinkChange et MRP_TopologyChange au niveau d'un port en anneau dont l'état est BLOCKED	Obligatoire
2 ... 65 535	Réservé	—

8.1.23 Codage du champ MRP_ManufacturerOUI

Ce champ doit être codé en tant que type de données octetString[3] avec l'OUI (Organizationally Unique Identifier) du fabricant de l'appareil, comme défini par le Registration Authority Committee (RAC) de l'IEEE.

8.1.24 Codage du champ MRP_IECOUI

Ce champ doit être codé en tant que type de données octetString[3] avec l'OUI (Organizationally Unique Identifier) de l'IEC, à savoir le 00-15-4e, comme défini par le Registration Authority Committee (RAC) de l'IEEE.

8.1.25 Codage du champ MRP_ManufacturerData

Ce champ doit être réservé aux données spécifiques au fournisseur.

8.1.26 Codage du champ MRP_DomainUUID

Ce champ doit être codé en UUID avec les valeurs conformes au Tableau 38.

Tableau 38 – MRP_DomainUUID

Valeur (hexadécimale)	Signification	Utilisation
0x00000000-0000-0000-0000-000000000000		Réservé
0x00000000-0000-0000-0000-000000000001 - 0xFFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFFFE	UUID du domaine de redondance du MRP	Facultative
0xFFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFFF	UUID par défaut du domaine de redondance du MRP	Obligatoire

8.1.27 Codage du champ MRP_InState

Ce champ doit être codé comme type de données Unsigned16 avec les valeurs conformes au Tableau 39.

Tableau 39 – MRP_InState

Valeur (hexadécimale)	Signification	Utilisation
0x0000	Interconnexion ouverte	Le MIM est à l'état de topologie d'interconnexion ouverte
0x0001	Interconnexion fermée	Le MIM est à l'état de topologie d'interconnexion fermée
0x0002 – 0xFFFF	Réservé	—

8.1.28 Codage du champ MRP_InID

Ce champ doit être codé en tant que type de données Unsigned16. Il est utilisé pour identifier le domaine d'interconnexion. La plage de valeurs va de 0 à 65 535. Le processus d'application demandeur doit fournir un MRP_InID unique à chaque topologie d'interconnexion.

8.2 Machines de protocole

8.2.1 Machine de protocole du MRM

La machine de protocole du MRM est définie au Tableau 41. Le comportement principal de la machine de protocole est présenté à la Figure 11.

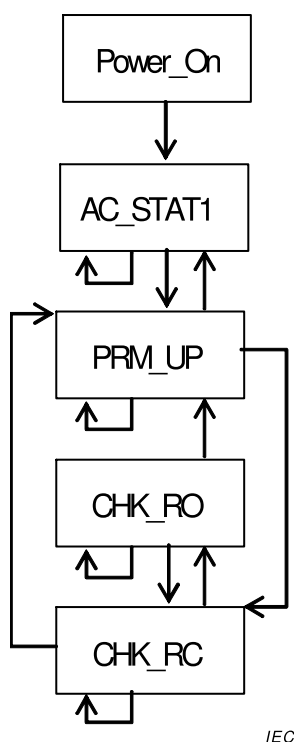


Figure 11 – Machine de protocole MRP pour le MRM

Le texte ci-dessous explique les principales actions réalisées dans les états. En cas de différence d'interprétation entre ce texte et le diagramme d'états, ce dernier l'emporte.

Power_On

Initialisation, le MRM doit démarrer les deux ports en anneau RPort_1 et RPort_2 à l'état BLOCKED. Les entrées statiques de la FDB pour les adresses multidiffusion MC_TEST, MC_CONTROL et MC_INCONTROL du MRP sont générées vers l'hôte. Tous les MRP-PDU doivent utiliser la priorité la plus élevée (ORG).

AC_STAT1 (état en attente de connexion 1)

Démarrage, attente du premier établissement de liaison au niveau de l'un de ses ports en anneau (appelé port en anneau principal), début de la surveillance d'essai de l'anneau et transition vers PRM_UP.

PRM_UP (Primary Ring Port with Link Up)

Cet état doit être atteint uniquement si le port en anneau principal comporte une liaison (port en anneau secondaire sans liaison). Le MRM doit régulièrement envoyer des trames MRP_Test par les deux ports en anneau même si l'autre port en anneau (port en anneau secondaire) n'a détecté aucune liaison.

CHK_RO (Check Ring, état Ring Open)

Le MRM n'a pas reçu ses trames MRP_Test pendant une durée déterminée, MRP_RingState doit être défini à l'état d'anneau ouvert.

Cet état peut également être atteint en cas de réception d'une trame MRP_LinkDown si l'option MRP_REACT_ON_LINK_CHANGE est prise en charge.

CHK_RC (Check Ring, état Ring Closed)

Le MRM doit envoyer ses trames MRP_Test et doit vérifier la liaison de ses ports en anneau, MRP_RingState doit être défini à l'état d'anneau fermé.

Les variables locales de la machine de protocole du MRM figurent au Tableau 40.

Tableau 40 – Variables locales du MRP de la machine de protocole du MRM

Nom	Type	Signification
SA_Port1	OctetString[6]	Adresse source MAC RPort_1 du port en anneau
SA_Port2	OctetString[6]	Adresse source MAC RPort_2 du port en anneau
SA_RPort	OctetString[6]	Adresse source MAC du port en anneau 1 ou 2
PRIORITY	Unsigned8	Priorité conforme à l'IEEE 802.1Q pour les MRP-PDU. Doit être définie sur ORG.
MRP_TS_Prio	Unsigned16	MRP_Prio de l'hôte
MRP_TS_SA	OctetString[6]	Adresse source MAC de l'hôte
RPort_1	Unsigned16	Identification de port du port en anneau 1
RPort_2	Unsigned16	Identification de port du port en anneau 2
PRM_RPort	Unsigned16	Identification de port du port en anneau principal
SEC_RPort	Unsigned16	Identification de port du port en anneau secondaire
MRP_MRM_NRmax	Unsigned16	Nombre maximal de retransmissions de MRP-PDU de type MRP_Test
MRP_MRM_NReturn	Unsigned16	Compteur, plage MRP_MRM_NRmax sur 0
TC_NReturn	Unsigned16	Compteur, plage MRP_TOPNRmax sur 0
ADD_TEST	Boolean	Envoyer un MRP-PDU supplémentaire de type MRP_Test après intervalle MRP_TSTshortT si la valeur est TRUE
REACT_MODE	Boolean	Le MRM réagit aux trames MRP_LinkDown à partir d'un MRC si la valeur est TRUE
MRP_LNK_UP	Unsigned16	Valeur constante pour indiquer un établissement de liaison
MRP_LNK_DOWN	Unsigned16	Valeur constante pour indiquer une interruption de liaison
MRP_BLOCKED_SUPPORTED	Unsigned16	Valeur constante pour indiquer que, si la valeur TRUE est affectée, le MRM prend pour hypothèse que tous les MRC de l'anneau prennent en charge l'état de port BLOCKED Si la valeur FALSE est affectée, le MRM exige une prise en charge supplémentaire pour les MRC ne prenant pas en charge l'état BLOCKED. FALSE: option pour les nœuds non conformes à l'IEC 61784-2, CP3/4, CP3/5, CP3/6.
MRP_REACT_ON_LINK_CHANGE	Unsigned16	Valeur constante pour indiquer que, si la valeur TRUE est affectée, le MRM réagit aux trames MRP_LinkDown provenant d'un MRC avec TopologyChange. Si la valeur FALSE est attribuée, le MRM ne réagit pas aux trames MRP_LinkDown. TRUE: option pour les nœuds non conformes à l'IEC 61784-1, IEC 61784-2, CP3/4, CP3/5, CP3/6
NO_TC	Boolean	Supprimer MRP_TopologyChange dans la topologie linéaire (NO_TC = TRUE)

Le diagramme d'états du MRM doit être conforme au Tableau 41.

Tableau 41 – Diagramme d'états du MRM

#	Etat en cours	Événement /Condition =>Action	Etat suivant
1	Power On	=> InitFDB() AddMACFDB({local},{MC_TEST, MC_CONTROL},ORG) AddMACFDB({local},{MC_INCONTROL},ORG) PRM_RPort:= RPort_1 SEC_RPort:= RPort_2 MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE REACT_MODE:= TRUE si le MRM réagit aux trames MRP_LinkDown provenant d'un MRC ou FALSE si le MRM ne réagit pas aux trames MRP_LinkDown provenant d'un MRC Set_Port_StateReq (PRM_RPort, BLOCKED) Set_Port_StateReq (SEC_RPort, BLOCKED)	AC_STAT1
2	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT)	PRM_UP
3	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => SEC_RPort:= PRM_RPort PRM_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT)	PRM_UP
5	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
6	AC_STAT1	Expiration de TestTimer => ignore	AC_STAT1

#	Etat en cours	Événement /Condition =>Action	Etat suivant
7	AC_STAT1	LinkChangeInd(PortMode, Link_status) => ignore	AC_STAT1
8	PRM_UP	Expiration de TestTimer => ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	PRM_UP
9	PRM_UP	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PRM_UP
10	PRM_UP	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => TestTimer.stop SetPortStateReq(PRM_RPort, BLOCKED)	AC_STAT1
11	PRM_UP	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	PRM_UP
12	PRM_UP	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= TRUE TestRingReq(MRP_TSTdefaultT)	CHK_RC
13	PRM_UP	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
14	PRM_UP	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA => ignore	PRM_UP

#	Etat en cours	Événement /Condition =>Action	Etat suivant
15	PRM_UP	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && PortMode == MRP_BLOCKED_SUPPORTED => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	PRM_UP
16	PRM_UP	LinkChangeInd(PortMode, Link_status) /ADD_TEST && PortMode == MRP_BLOCKED_SUPPORTED => ignore	PRM_UP
17	PRM_UP	LinkChangeInd(PortMode, Link_status) /Link_status == MRP_LNK_DOWN && PortMode != MRP_BLOCKED_SUPPORTED => ignore	PRM_UP
18	PRM_UP	LinkChangeInd(PortMode, Link_status) /ADD_TEST && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => TopologyChangeReq(0)	PRM_UP
19	PRM_UP	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT) TopologyChangeReq(0)	PRM_UP
20	PRM_UP	TopologyChangeInd(MRP_SA, t) => ignore	PRM_UP
21	CHK_RO	Expiration de TestTimer => ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RO
22	CHK_RO	MAUTypeChangeInd(RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RO

#	Etat en cours	Événement /Condition =>Action	Etat suivant
23	CHK_RO	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	PRM_UP
24	CHK_RO	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => Ignore	CHK_RO
25	CHK_RO	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq (SEC_RPort, BLOCKED)	PRM_UP
26	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA && REACT_MODE != MRP_REACT_ON_LINK_CHANGE => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	CHK_RC
27	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA && REACT_MODE == MRP_REACT_ON_LINK_CHANGE => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(0)	CHK_RC
28	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA => ignore	CHK_RO

#	Etat en cours	Événement /Condition =>Action	Etat suivant
29	CHK_RO	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && Link_status == MRP_LNK_UP && PortMode == MRP_BLOCKED_SUPPORTED => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RO
30	CHK_RO	LinkChangeInd(PortMode, Link_status) /ADD_TEST && Link_status == MRP_LNK_UP && PortMode == MRP_BLOCKED_SUPPORTED => Ignore	CHK_RO
31	CHK_RO	LinkChangeInd(PortMode, Link_status) /ADD_TEST && Link_status == MRP_LNK_DOWN => Ignore	CHK_RO
32	CHK_RO	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && Link_status == MRP_LNK_DOWN => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RO
33	CHK_RO	LinkChangeInd(PortMode, Link_status) /ADD_TEST && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTExtNRmax – 1 MRP_MRM_NReturn:= 0 TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(0)	CHK_RC

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
34	CHK_RO	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTExtNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT) TopologyChangeReq(0)	CHK_RC
35	CHK_RO	TopologyChangeInd(MRP_SA, t) => ignore	CHK_RO
36	CHK_RC	Expiration de TestTimer /MRP_MRM_NReturn = MPI_MPM_NPμξ && !NO_TC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE TopologyChangeReq(MRP_TOPchgT) TestRingReq(MRP_TSTdefaultT)	CHK_RO
37	CHK_RC	Expiration de TestTimer /MRP_MRM_NReturn = MPI_MPM_NPμξ && NO_TC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RO
38	CHK_RC	Expiration de TestTimer /MRP_MRM_NReturn MPI_MPM_NPμξ => MRP_MRM_NReturn:= MRP_MRM_NReturn 1 ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
39	CHK_RC	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Ignore	CHK_RC

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
40	CHK_RC	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	PRM_UP
41	CHK_RC	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => Ignore	CHK_RC
42	CHK_RC	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => Ignore	PRM_UP
43	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE	CHK_RC
44	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA => ignore	CHK_RC
45	CHK_RC	LinkChangeInd(PortMode, Link_status) /ADD_TEST && REACT_MODE != MRP_REACT_ON_LINK_CHANGE && PortMode == MRP_BLOCKED_SUPPORTED => ignore	CHK_RC
46	CHK_RC	LinkChangeInd(PortMode, Link_status) /!ADD_TEST && REACT_MODE != MRP_REACT_ON_LINK_CHANGE && PortMode == MRP_BLOCKED_SUPPORTED => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RC

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
47	CHK_RC	LinkChangeInd(PortMode, Link_status) /Link_status == MRP_LNK_DOWN && REACT_MODE == MRP_REACT_ON_LINK_CHANGE => Set_Port_StateReq (SEC_RPort, FORWARDING) TopologyChangeReq(0)	CHK_RO
48	CHK_RC	LinkChangeInd(PortMode, Link_status) /Link_status == MRP_LNK_UP && REACT_MODE == MRP_REACT_ON_LINK_CHANGE && PortMode != MRP_BLOCKED_SUPPORTED => MRP_MRM_NRmax:= MRP_TSTExtNRmax 1 TopologyChangeReq(0)	CHK_RC
49	CHK_RC	LinkChangeInd(PortMode, Link_status) /Link_status == MRP_LNK_UP && REACT_MODE == MRP_REACT_ON_LINK_CHANGE && PortMode == MRP_BLOCKED_SUPPORTED => MRP_MRM_NRmax:= MRP_TSTNRmax 1 TopologyChangeReq(0)	CHK_RC
50	CHK_RC	TopologyChangeInd(MRP_SA, t) => Ignore	CHK_RC
51	PRM_UP	InterconnTopologyChangeInd(MRP_SA, t, InID) /TopTimer.isRunning == FALSE => TopologyChangeReq(t)	PRM_UP
52	CHK_RO	InterconnTopologyChangeInd(MRP_SA, t, InID) /TopTimer.isRunning == FALSE => TopologyChangeReq(t)	CHK_RO
53	CHK_RO	InterconnLinkChangeInd(InID, LinkStatus, RPort) /RPort == PRM_RPort => InterconnForwardReq(SEC_RPort)	CHK_RO
54	CHK_RO	InterconnLinkChangeInd(InID, LinkStatus, RPort) /RPort == SEC_RPort => InterconnForwardReq(PRM_RPort)	CHK_RO
55	CHK_RO	InterconnLinkStatusInd(InID, RPort) /RPort == PRM_RPort >= InterconnForwardReq(SEC_RPort)	CHK_RO

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
56	CHK_RO	InterconnLinkStatusInd(InID, RPort) /RPort == SEC_RPort => InterconnForwardReq(PRM_RPort)	CHK_RO
57	CHK_RC	InterconnTopologyChangeInd(MRP_SA, t, InID) /TopTimer.isRunning == FALSE => TopologyChangeReq(t)	CHK_RC

8.2.2 Machine de protocole du MRC

La machine de protocole du MRC est définie au Tableau 43. Le comportement principal de la machine de protocole est présenté à la Figure 12.

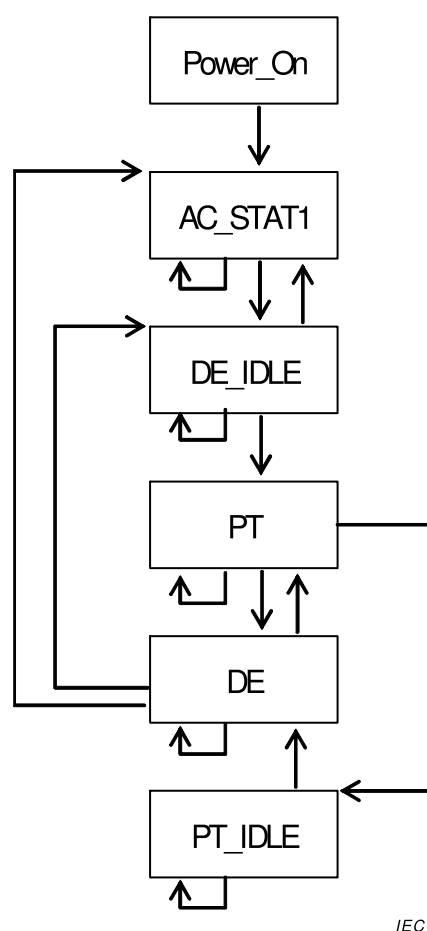


Figure 12 – Machine de protocole MRP pour le MRC

Le texte ci-dessous explique les principales actions réalisées dans les états. En cas de différence d'interprétation entre ce texte et le diagramme d'états, ce dernier l'emporte.

Power_On

Initialisation, le MRC doit démarrer les deux ports en anneau RPort_1 et RPort_2 à l'état BLOCKED. Les entrées statiques de la FDB pour les adresses multidiffusion MC_TEST et MC_CONTROL du MRP sont générées: Transférer les trames du MRP vers MC_TEST et MC_CONTROL entre les ports en anneau, ainsi que les trames

MC_CONTROL vers l'hôte. Tous les MRP-PDU doivent utiliser la priorité la plus élevée (ORG).

AC_STAT1 (état en attente de connexion 1)

Démarrage, attente de l'établissement de liaison au niveau de l'un de ses ports en anneau.

DE_IDLE (état Data Exchange Idle)

Cet état doit être atteint si un seul port en anneau (principal) comporte une liaison et que l'état de son port est FORWARDING.

PT (transmission)

Etat temporaire lors du signalement des changements de liaison.

DE (Data Exchange)

Etat temporaire lors du signalement des changements de liaison.

PT_IDLE (état Pass Through Idle)

Cet état doit être atteint si les deux ports en anneau comportent une liaison et que les états du port sont FORWARDING.

Les variables locales de la machine de protocole du MRC figurent au Tableau 42.

Tableau 42 – Variables locales du MRP de la machine de protocole du MRC

Nom	Type	Signification
SA_RPort	OctetString[6]	Adresse source MAC du port en anneau 1 ou 2
PRIORITY	Unsigned8	Priorité conforme à l'IEEE 802.1Q pour les MRP-PDU. Doit être définie sur ORG.
RPort_1	Unsigned16	Identification de port du port en anneau 1
RPort_2	Unsigned16	Identification de port du port en anneau 2
PRM_RPort	Unsigned16	Identification de port du port en anneau principal
SEC_RPort	Unsigned16	Identification de port du port en anneau secondaire
MRP_LNKNReturn	Unsigned16	Compteur, plage MRP_LNKNRmax sur 0
MRP_LNK_UP	Unsigned16	Valeur constante pour indiquer un établissement de liaison
MRP_LNK_DOWN	Unsigned16	Valeur constante pour indiquer une interruption de liaison

Le diagramme d'états du MRC doit être conforme au Tableau 43.

Tableau 43 – Diagramme d'états du MRC

#	Etat en cours	Événement /Condition =>Action	Etat suivant
1	Power On	=> InitFDB() AddMACFDB({RPort_1,RPort_2}, {MC_TEST, MC_CONTROL},ORG) AddMACFDB({local},{MC_CONTROL},ORG) PRM_RPort:= RPort_1 SEC_RPort:= RPort_2 Set_Port_StateReq (PRM_RPort, BLOCKED) Set_Port_StateReq (SEC_RPort, BLOCKED) UpTimer.ini DownTimer.ini	AC_STAT1
2	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Set_Port_StateReq (PRM_RPort, FORWARDING)	DE_IDLE
3	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => SEC_RPort:= PRM_RPort PRM_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING)	DE_IDLE
5	AC_STAT1	TopologyChangeInd(MRP_SA, t) => ignore	AC_STAT1
6	DE_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.start(MRP_LNKupT) LinkChangeReq(PRM_RPort, MRP_LNK_UP, MRP_LNKNReturn X MRP_LNKupT)	PT

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
7	DE_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE_IDLE
8	DE_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq (PRM_RPort, BLOCKED)	AC_STAT1
9	DE_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE_IDLE
10	DE_IDLE	TopologyChangeInd(MRP_SA, t) => ClearFDB(t)	DE_IDLE
11	PT	UpTimer expired /MRP_LNKNReturn == 0 => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (SEC_RPort, FORWARDING)	PT_IDLE
12	PT	UpTimer expired /MRP_LNKNReturn 0 => MRP_LNKNReturn:= MRP_LNKNReturn + 1 UpTimer.start(MRP_LNKupT) LinkChangeReq(PRM_RPort, MRP_LNK_UP, MRP_LNKNReturn X MRP_LNKupT)	PT
13	PT	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT

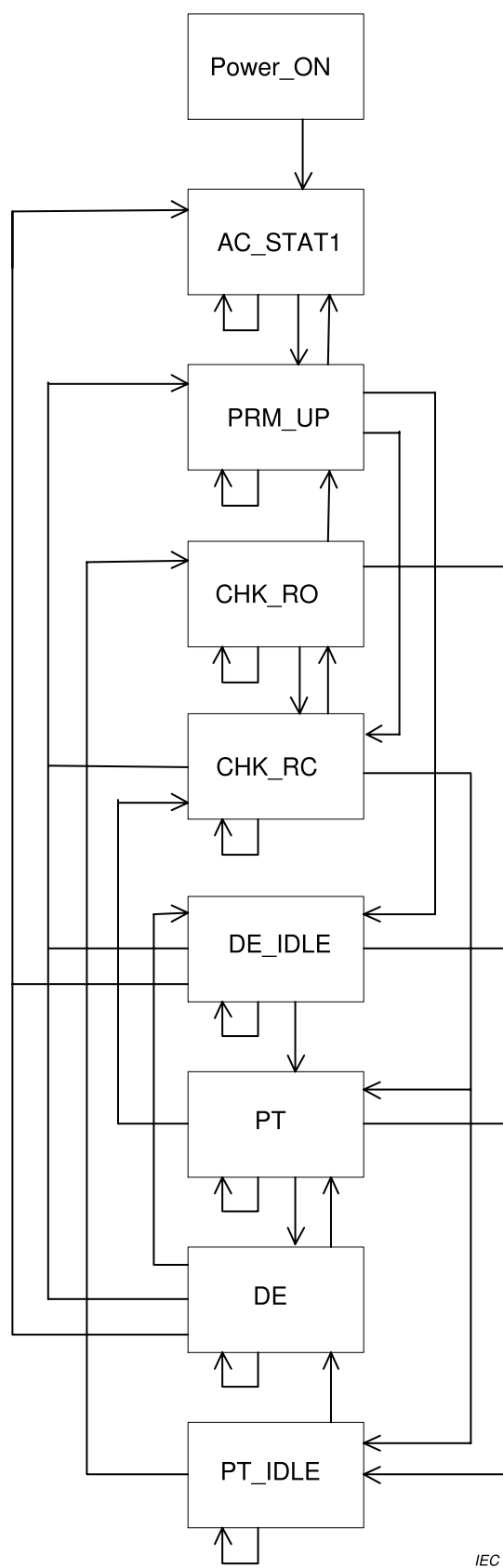
#	Etat en cours	Événement /Condition =>Action	Etat suivant
14	PT	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKReturn:= MRP_LNKmax UpTimer.stop Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort , MRP_LNK_DOWN, MRP_LNKReturn X MRP_LNKdownT)	DE
15	PT	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKReturn:= MRP_LNKmax UpTimer.stop PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING) Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort , MRP_LNK_DOWN, MRP_LNKReturn X MRP_LNKdownT)	DE
16	PT	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT
17	PT	TopologyChangeInd(MRP_SA, t) => MRP_LNKReturn:= MRP_LNKmax UpTimer.stop Set_Port_StateReq (SEC_RPort, FORWARDING) ClearFDB(t)	PT_IDLE
18	DE	DownTimer expired /MRP_LNKReturn == 0 => MRP_LNKReturn:= MRP_LNKmax	DE_IDLE

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
19	DE	DownTimer expired /MRP_LNKNReturn 0 => MRP_LNKNReturn:= MRP_LNKNReturn – 1 DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn X MRP_LNKdownT)	DE
20	DE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_LNKNReturn:= MRP_LNKNRmax DownTimer.stop UpTimer.start(MRP_LNKupT) LinkChangeReq(PRM_RPort , MRP_LNK_UP, MRP_LNKNReturn X MRP_LNKupT)	PT
21	DE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE
22	DE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (PRM_RPort, BLOCKED) DownTimer.stop	AC_STAT1
23	DE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE
24	DE	TopologyChangeInd(MRP_SA, t) => MRP_LNKNReturn:= MRP_LNKNRmax DownTimer.stop ClearFDB(t)	DE_IDLE

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
25	PT_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
26	PT_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKReturn:= MRP_LNKmax Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort , MRP_LNK_DOWN, MRP_LNKReturn X MRP_LNKdownT)	DE
27	PT_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKReturn:= MRP_LNKmax PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort , MRP_LNK_DOWN, MRP_LNKReturn X MRP_LNKdownT)	DE
28	PT_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
29	PT_IDLE	TopologyChangeInd(MRP_SA, t) => ClearFDB(t)	PT_IDLE

8.2.3 Machine de protocole du MRA

La machine de protocole du MRA est définie au Tableau 45. Le comportement principal de la machine de protocole est présenté à la Figure 13.



IEC

Figure 13 – Machine de protocole MRP pour le MRA

La machine de protocole du MRA consiste essentiellement en une combinaison de la machine du protocole du MRM et de celle du MRC.

Pour des explications sur les états de la machine de protocole du MRA, voir les explications relatives à la machine de protocole du MRM en 8.2.1 et celles sur le diagramme d'états du MRC en 8.2.2. En cas de différence d'interprétation entre les textes d'explications et le diagramme d'états, ce dernier l'emporte.

La machine de protocole du MRA utilise les variables locales des machines du protocole du MRM et du MRC. En outre, la machine de protocole du MRA utilise d'autres variables locales, comme l'indique le Tableau 44.

Tableau 44 – Variables locales du MRP de la machine de protocole du MRA

Nom	Type	Signification
MRP_BestMRM_SA	OctetString[6]	Adresse source MAC du gestionnaire enregistré dont la priorité est actuellement la plus élevée
MRP_BestMRM_Prio	Unsigned16	MRP_Prio du gestionnaire enregistré dont la priorité est actuellement la plus élevée
HO_BestMRM_SA	OctetString[6]	Adresse source MAC du gestionnaire enregistré dont la priorité est la plus élevée stockée par l'hôte
HO_BestMRM_Prio	Unsigned16	MRP_Prio du gestionnaire enregistré dont la priorité est la plus élevée stockée par l'hôte
MRP_MonNRReturn	Unsigned16	Compteur de nouvelles tentatives pour surveiller le MRA enregistré dont la priorité est la plus élevée
MRP_mySA	OctetString[6]	Adresse source MAC hôte
MRP_myPrio	Unsigned16	Priorité de l'hôte
MRP_remotePrio	Unsigned16	Priorité reçue
LNKUP_HYST_TIMER_RUNNING	Unsigned16	Drapeau pour le temporisateur d'hystérésis d'établissement de liaison: TRUE: Le temporisateur est en cours d'exécution FALSE: Le délai indiqué s'est écoulé ou le temporisateur n'a pas encore été démarré. La maintenance du temporisateur d'hystérésis d'établissement de liaison est réalisée hors du diagramme d'états. Il doit être démarré lorsque l'appareil reconnaît un établissement de liaison et doit être arrêté lorsque le délai indiqué par le temporisateur s'est écoulé ou que l'appareil reconnaît une interruption de liaison au niveau du port. A la reconnaissance de l'établissement de liaison, MAUType_ChangeInd doit d'abord être lancé, lorsque le délai indiqué par le temporisateur d'hystérésis d'établissement de liaison s'est écoulé.

Le diagramme d'états du MRA doit être conforme au Tableau 45.

Tableau 45 – Diagramme d'états du MRA

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
1	Power On	=> InitFDB() AddMACFDB({local},{MC_TEST, MC_CONTROL},ORG) PRM_RPort:= RPort_1 SEC_RPort:= RPort_2 MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE REACT_MODE:= FALSE MRP_BestMRM_SA:= 0xFFFFFFFFFFFF MRP_BestMRM_Prio:= 0xFFFF Set_Port_StateReq (PRM_RPort, BLOCKED) Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MonNReturn:= 0	AC_STAT1
2	AC_STAT1	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT)	PRM_UP
3	AC_STAT1	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
4	AC_STAT1	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => SEC_RPort:= PRM_RPort PRM_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT)	PRM_UP
5	AC_STAT1	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	AC_STAT1
6	AC_STAT1	TestTimer expired => ignore	AC_STAT1
7	AC_STAT1	LinkChangeInd(PortMode, Link_status) => ignore	AC_STAT1
8	PRM_UP	TestTimer expired => ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	PRM_UP

#	Etat en cours	Événement /Condition =>Action	Etat suivant
9	PRM_UP	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status ==MRP_LNK_UP => ignore	PRM_UP
10	PRM_UP	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => TestTimer.stop SetPortStateReq(PRM_RPort, BLOCKED)	AC_STAT1
11	PRM_UP	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort&& Link_status ==MRP_LNK_DOWN => ignore	PRM_UP
12	PRM_UP	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= TRUE TestRingReq(MRP_TSTdefaultT)	CHK_RC
13	PRM_UP	TestRingInd (MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
14	PRM_UP	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => ignore	PRM_UP
15	PRM_UP	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => TestMgrNAckReq(MRP_Prio, MRP_SA)	PRM_UP
16	PRM_UP	LinkChangeInd(PortMode, Link_status) /!ADD_TEST => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	PRM_UP
17	PRM_UP	LinkChangeInd(PortMode, Link_status) /ADD_TEST => ignore	PRM_UP

#	Etat en cours	Événement /Condition =>Action	Etat suivant
18	PRM_UP	TopologyChangeInd(MRP_SA, t) /MRP_SA == MRP_TS_SA => ignore	PRM_UP
19	PRM_UP	TopologyChangeInd(MRP_SA, t) /MRP_SA != MRP_TS_SA => CLEAR_FDB(t)	PRM_UP
20	PRM_UP	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA == MRP_TS_SA => ignore	PRM_UP
21	PRM_UP	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA != MRP_TS_SA => ignore	PRM_UP
22	PRM_UP	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA ==MRP_TS_SA && !REM_PRIO_BETTER_THAN_HO_PRIO => TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	DE_IDLE
23	PRM_UP	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && REM_PRIO_BETTER_THAN_HO_PRIO => HO_BestMRM_SA:= MRP_SA HO_BestMRM_Prio:= MRP_Prio TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	DE_IDLE
24	PRM_UP	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) => ignore	PRM_UP
25	CHK_RO	TestTimer expired => ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RO
26	CHK_RO	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_Rport && Link_status == MRP_LNK_UP => ignore	CHK_RO

#	Etat en cours	Événement /Condition =>Action	Etat suivant
27	CHK_RO	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => PRM_RPort:= SEC_RPort SEC_RPort:= Rport Set_Port_StateReq(SEC_RPort, BLOCKED) TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	PRM_UP
28	CHK_RO	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_Rport && Link_status == MRP_LNK_UP => ignore	CHK_RO
29	CHK_RO	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq(SEC_RPort, BLOCKED)	PRM_UP
30	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	CHK_RC
31	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => ignore	CHK_RO
32	CHK_RO	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => TestMgrNAckReq(MRP_Prio, MRP_SA)	CHK_RO
33	CHK_RO	LinkChangeInd(PortMode, LinkStatus) /!ADD_TEST => ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RO
34	CHK_RO	LinkChangeInd(PortMode, LinkStatus) /ADD_TEST => ignore	CHK_RO

#	Etat en cours	Événement /Condition =>Action	Etat suivant
35	CHK_RO	TopologyChangeInd(MRP_SA, t) /MRP_SA == MRP_TS_SA => ignore	CHK_RO
36	CHK_RO	TopologyChangeInd(MRP_SA, t) /MRP_SA != MRP_TS_SA => CLEAR_FDB(t)	CHK_RO
37	CHK_RO	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA == MRP_TS_SA => ignore	CHK_RO
38	CHK_RO	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA != MRP_TS_SA => ignore	CHK_RO
39	CHK_RO	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && !REM_PRIO_BETTER_THAN_HO_PRIO => TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	PT_IDLE
40	CHK_RO	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && REM_PRIO_BETTER_THAN_HO_PRIO => O_BestMRM_SA:= MRP_SA HO_BestMRM_Prio:= MRP_Prio TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	PT_IDLE
41	CHK_RO	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) => ignore	CHK_RO
42	CHK_RC	TestTimer expired /MRP_MRM_NReturn >= MRP_MRM_Nrmax && !NO_TC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE TopologyChangeReq(MRP_TOPchgT) TestRingReq(MRP_TSTdefaultT)	CHK_RO

#	Etat en cours	Événement /Condition =>Action	Etat suivant
43	CHK_RC	TestTimer expired /MRP_MRM_NReturn >= MRP_MRM_Nrmax && NO_TC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_Nrmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 ADD_TEST:= FALSE TopologyChangeReq(MRP_TOPchgT) TestRingReq(MRP_TSTdefaultT)	CHK_RO
44	CHK_RC	TestTimer expired /MRP_MRM_NReturn < MRP_MRM_Nrmax => MRP_MRM_NReturn:= MRP_MRM_NReturn + 1 ADD_TEST:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
45	CHK_RC	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RC
46	CHK_RC	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) Set_Port_StateReq (PRM_RPort, FORWARDING) TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	PRM_UP
47	CHK_RC	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RC
48	CHK_RC	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	PRM_UP
49	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => MRP_MRM_Nrmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NO_TC:= FALSE	CHK_RC
50	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => TestMgrNAckReq(MRP_Prio, MRP_SA)	CHK_RC

#	Etat en cours	Événement /Condition =>Action	Etat suivant
51	CHK_RC	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => ignore	CHK_RC
52	CHK_RC	LinkChangeInd(PortMode, Link_status) /ADD_TEST => ignore	CHK_RC
53	CHK_RC	LinkChangeInd(PortMode, Link_status) /!ADD_TEST =>ADD_TEST:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RC
54	CHK_RC	TopologyChangeInd(MRP_SA, t) /MRP_SA == MRP_TS_SA => ignore	CHK_RC
55	CHK_RC	TopologyChangeInd(MRP_SA, t) /MRP_SA != MRP_TS_SA => CLEAR_FDB(t)	CHK_RC
56	CHK_RC	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA == MRP_TS_SA => ignore	CHK_RC
57	CHK_RC	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA != MRP_TS_SA => ignore	CHK_RC
58	CHK_RC	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && !REM_PRIO_BETTER_THAN_HO_PRIO => Set_Port_StateReq (SEC_RPort, FORWARDING) TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	PT_IDLE
59	CHK_RC	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) /MRP_SA != MRP_TS_SA && MRP_BestMRM_SA == MRP_TS_SA && REM_PRIO_BETTER_THAN_HO_PRIO => HO_BestMRM_SA:= MRP_SA HO_BestMRM_Prio:= MRP_Prio TopTimer.stop MRC_Init() TestPropagateReq(HO_BestMRM_Prio, HO_BestMRM_SA)	PT_IDLE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
60	CHK_RC	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) => ignore	CHK_RC
61	DE_IDLE	TestTimer expired /MRP_MonNReturn <= MRP_MON_NR_MAX => MRP_MonNReturn:= MRP_MonNReturn + 1 TestTimer.start(MRP_TSTshortT)	DE_IDLE
62	DE_IDLE	TestTimer expired /MRP_MonNReturn > MRP_MON_NR_MAX => TestTimer.start(MRP_TSTshortT) MRM_Init()	PRM_UP
63	DE_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.start(MRP_LNKupT) AddMACFDB({PRM_RPort, SEC_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_Rport, MRP_LNK_UP, MRP_LNKNReturn * MRP_LNKupT)	PT
64	DE_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE_IDLE
65	DE_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq (PRM_RPort, BLOCKED) AddMACFDB({}, {MC_TEST, MC_CONTROL},ORG)	AC_STAT1
66	DE_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE_IDLE
67	DE_IDLE	TopologyChangeInd(MRP_SA, t) => CLEAR_FDB(t)	DE_IDLE
68	DE_IDLE	TopologyChangeInd(MRP_SA, t) /LNKUP_HYST_TIMER_RUNNING == TRUE => CLEAR_FDB(t) AddMACFDB({PRM_RPort,SEC_RPort}, {MC_TEST, MC_CONTROL},ORG) Set_Port_StateReq (SEC_RPort,FORWARDING)	PT_IDLE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
69	DE_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => ignore	DE_IDLE
70	DE_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	DE_IDLE
71	DE_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA &&MRP_SA == HO_BestMRM_SA &&!REM_PRIO_BETTER_THAN_OWN_PRIO => HO_BestMRM_Prio:= MRP_Prio	DE_IDLE
72	DE_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA &&MRP_SA == HO_BestMRM_SA &&REM_PRIO_BETTER_THAN_OWN_PRIO => MRP_MonNReturn:= 0 HO_BestMRM_Prio:= MRP_Prio	DE_IDLE
73	DE_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA == MRP_TS_SA => ignore	DE_IDLE
74	DE_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	DE_IDLE
75	DE_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == MRP_BestMRM_SA => HO_BestMRM_SA:= MRP_BestMRM_SA HO_BestMRM_Prio:= MRP_BestMRM_Prio MRP_MonNReturn:= 0	DE_IDLE
76	DE_IDLE	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) => ignore	DE_IDLE
77	DE_IDLE	LinkChangeInd(PortMode, Link_status) => ignore	DE_IDLE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
78	PT	TestTimer expired /MRP_MonNReturn <= MRP_MON_NR_MAX => MRP_MonNReturn:= MRP_MonNReturn + 1 TestTimer.start(MRP_TSTshortT)	PT
79	PT	TestTimer expired /MRP_MonNReturn > MRP_MON_NR_MAX => TestTimer.start(MRP_TSTshortT) MRM_Init()	CHK_RC
80	PT	UpTimer expired /MRP_LNKNReturn == 0 => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (SEC_RPort, FORWARDING)	PT_IDLE
81	PT	UpTimer expired /MRP_LNKNReturn > 0 => MRP_LNKNReturn:= MRP_LNKNReturn - 1 UpTimer.start(MRP_LNKupT) LinkChangeReq(PRM_RPort, MRP_LNK_UP, MRP_LNKNReturn * MRP_LNKupT)	PT
82	PT	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT
83	PT	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.stop Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) AddMACFDB({PRM_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE
84	PT	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.stop PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (PRM_RPort, FORWARDING) Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) AddMACFDB({PRM_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
85	PT	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT
86	PT	TopologyChangeInd(MRP_SA, t) => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.stop Set_Port_StateReq (SEC_RPort, FORWARDING) CLEAR_FDB(t)	PT_IDLE
87	PT	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => ignore	PT
88	PT	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	PT
89	PT	TestRingInd(MRP_SA, MRP_Prio /MRP_SA != MRP_TS_SA && MRP_SA == HO_BestMRM_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => HO_BestMRM_Prio:= MRP_Prio	PT
90	PT	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == HO_BestMRM_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => MRP_MonNReturn:= 0 HO_BestMRM_Prio:= MRP_Prio	PT
91	PT	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA == MRP_TS_SA => ignore	PT
92	PT	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	PT

#	Etat en cours	Événement /Condition =>Action	Etat suivant
93	PT	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == MRP_BestMRM_SA => HO_BestMRM_SA:= MRP_BestMRM_SA HO_BestMRM_Prio:= MRP_BestMRM_Prio MRP_MonNReturn:= 0	PT
94	PT	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) => ignore	PT
95	PT	LinkChangeInd(PortMode, Link_status) => ignore	PT
96	DE	TestTimer expired /MRP_MonNReturn <= MRP_MON_NR_MAX => MRP_MonNReturn:= MRP_MonNReturn + 1 TestTimer.start(MRP_TSTdefaultT)	DE
97	DE	TestTimer expired /MRP_MonNReturn > MRP_MON_NR_MAX => TestTimer.start(MRP_TSTshortT) MRM_Init()	PRM_UP
98	DE	DownTimer expired /MRP_LNKNReturn == 0 => MRP_LNKNReturn:= MRP_LNKNRmax	DE_IDLE
99	DE	DownTimer expired /MRP_LNKNReturn > 0 => MRP_LNKNReturn:= MRP_LNKNReturn – 1 DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE
100	DE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => MRP_LNKNReturn:= MRP_LNKNRmax DownTimer.stop UpTimer.start(MRP_LNKupT) AddMACFDB({PRM_RPort, SEC_RPort}, {MC_TEST, MC_CONTROL},ORG LinkChangeReq(PRM_RPort, MRP_LNK_UP, MRP_LNKNReturn * MRP_LNKupT)	PT
101	DE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
102	DE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (PRM_RPort, BLOCKED) DownTimer.stop AddMACFDB({}, {MC_TEST, MC_CONTROL},ORG)	AC_STAT1
103	DE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE
104	DE	TopologyChangeInd(MRP_SA, t) => MRP_LNKNReturn:= MRP_LNKNRmax DownTimer.stop CLEAR_FDB(t)	DE_IDLE
105	DE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => ignore	DE
106	DE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	DE
107	DE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == HO_BestMRM_SA && !REM_PRIO_BETTER_THAN_OWN_PRIO => HO_BestMRM_Prio:= MRP_Prio	DE
108	DE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == HO_BestMRM_SA && REM_PRIO_BETTER_THAN_OWN_PRIO => MRP_MonNReturn:= 0 HO_BestMRM_Prio:= MRP_Prio	DE
109	DE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA == MRP_TS_SA => ignore	DE
110	DE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	DE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
111	DE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == MRP_BestMRM_SA => HO_BestMRM_SA:= MRP_BestMRM_SA HO_BestMRM_Prio:= MRP_BestMRM_Prio MRP_MonNRReturn:= 0	DE
112	DE	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) => ignore	DE
113	DE	LinkChangeInd(PortMode, Link_status) => ignore	DE
114	PT_IDLE	TestTimer expired /MRP_MonNRReturn <= MRP_MON_NR_MAX => MRP_MonNRReturn:= MRP_MonNRReturn + 1 TestTimer.start(MRP_TSTdefaultT)	PT_IDLE
115	PT_IDLE	TestTimer expired /MRP_MonNRReturn > MRP_MON_NR_MAX => TestTimer.start(MRP_TSTshortT) MRM_Init()	CHK_RO
116	PT_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
117	PT_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) AddMACFDB({PRM_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
118	PT_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => MRP_LNKNReturn:= MRP_LNKNRmax PRM_RPort:= SEC_RPort SEC_RPort:= RPort Set_Port_StateReq (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) AddMACFDB({PRM_RPort}, {MC_TEST, MC_CONTROL},ORG) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn * MRP_LNKdownT)	DE
119	PT_IDLE	MAUType_ChangeInd (RPort, Link_status) /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
120	PT_IDLE	TopologyChangeInd(MRP_SA, t) => CLEAR_FDB(t)	PT_IDLE
121	PT_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA == MRP_TS_SA => ignore	PT_IDLE
122	PT_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	PT_IDLE
123	PT_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA &&MRP_SA == HO_BestMRM_SA &&!REM_PRIO_BETTER_THAN_OWN_PRIO => HO_BestMRM_Prio:= MRP_Prio	PT_IDLE
124	PT_IDLE	TestRingInd(MRP_SA, MRP_Prio) /MRP_SA != MRP_TS_SA MRP_SA == HO_BestMRM_SA REM_PRIO_BETTER_THAN_OWN_PRIO => MRP_MonNReturn:= 0 HO_BestMRM_Prio:= MRP_Prio	PT_IDLE
125	PT_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA == MRP_TS_SA => ignore	PT_IDLE

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
126	PT_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA != MRP_BestMRM_SA => ignore	PT_IDLE
127	PT_IDLE	TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio) /MRP_SA != MRP_TS_SA && MRP_SA == MRP_BestMRM_SA => HO_BestMRM_SA:= MRP_BestMRM_SA HO_BestMRM_Prio:= MRP_BestMRM_Prio MRP_MonNReturn:= 0	PT_IDLE
128	PT_IDLE	TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA) => ignore	PT_IDLE
129	PT_IDLE	LinkChangeInd(PortMode, Link_status) => ignore	PT_IDLE

8.2.4 Fonctions des MRA, MRM et MRC

Les fonctions des MRA, MRM et MRC doivent être conformes au Tableau 46.

Tableau 46 – Fonctions et macros du MRP

Nom de la fonction	Opérations
TestRingReq(t)	SetupTestRingReq() TestTimer.start(t)
SetupTestRingReq()	Créer un MRP-PDU conformément à MRP_Test Affectations: MRP_Type:= MRP_Test MRP_Prio:= MRP_TS_Prio MRP_SA:= MRP_TS_SA MRP_PortRole:= trame envoyée sur le port en anneau principal ou secondaire MRP_RingState:= état en cours de l'anneau MRP_Transition:= nombre réel de transitions entre les états d'anneau ouvert et fermé MRP_TimeStamp:= valeur en cours du compteur local MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID Si MRA MRP_Suboption1 = vendorspecific MRP_Suboption2 = MRP_AutoMgr MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_TEST, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_TEST, SA_Port2, PRIORITY, LT, MRP-PDU)
TestRingInd(MRP_SA, MRP_Prio)	Recevoir un MRP-PDU conformément à MRP_Test MRP_SA:= MRP_SA de MRP-PDU MRP_Prio:= MRP_Prio de MRP-PDU
TopologyChangeReq(time)	SetupTopologyChangeReq(MRP_TOPNRmax X time) si time == 0 ClearLocalFDB() sinon TopTimer.start(MRP_TOPchgT) TopTimer.isRunning = TRUE
SetupTopologyChangeReq(t)	Créer un MRP-PDU conformément à MRP_TopologyChange Affectations: MRP_Type:= MRP_TopologyChange MRP_Prio:= MRP_TS_Prio MRP_SA:= MRP_TS_SA MRP_Interval:= t MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_CONTROL, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_CONTROL, SA_Port2, PRIORITY, LT, MRP-PDU)

Nom de la fonction	Opérations
TopologyChangeInd(MRP_SA, t)	Recevoir un MRP-PDU conformément à MRP_TopologyChange MRP_SA:= MRP_SA de MRP_PDU t:= MRP_Interval de MRP-PDU
LinkChangeReq(RPort, LinkStatus, time)	Créer un MRP-PDU conformément à MRP_LinkUp ou MRP_LinkDown Affectations: si LinkStatus == MRP_LNK_UP MRP_Type:= MRP_LinkUp sinon MRP_Type:= MRP_LinkDown MRP_SA:= MRP_TS_SA MRP_PortRole:= rôle en cours du port qui a signalé le changement de liaison MRP_Interval:= time MRP_Blocked:= indique si le MRC peut ou non recevoir et transférer les trames MRP_Test, MRP_LinkChange et MRP_TopologyChange sur un port en anneau dont l'état est BLOCKED. MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort, MC_CONTROL, SA_RPort, PRIORITY, LT, MRP-PDU)
LinkChangeInd(PortMode, LinkStatus)	Recevoir un MRP-PDU conformément à MRP_LinkDown ou MRP_LinkUp PortMode:= MRP_Blocked de MRP-PDU si MRP_Type == MRP_LinkUp LinkStatus:= MRP_LNK_UP sinon LinkStatus:= MRP_LNK_DOWN
MAUTypeChangeInd(RPort, Link_status)	Recevoir une indication de changement de liaison locale. RPort:= port à l'origine de l'indication de changement de liaison locale. Link_status:= MRP_LNK_UP ou MRP_LNK_DOWN (en fonction de l'indication de changement de liaison locale)
SetPortStateReq(RPort, Status)	Fonction qui permet de configurer l'état de port de RPort sur Status
ClearFDB(time)	FDBCleartimer.start(time)
ClearLocalFDB()	Fonction qui permet d'annuler la FDB à l'intérieur du nœud du MRP. L'apprentissage des adresses sources à partir des trames entrantes, qui sont envoyées avant le signalement du changement de topologie, doit être évité
InitFDB()	Fonction d'initialisation de la base de données de filtrage
AddMACFDB(Destination, MAC address, Priority)	Fonction d'ajout d'entrées de filtrage statiques (adresse MAC) dans la FDB avec la priorité et la destination. Dans le schéma d'état, le terme "local" signifie une connexion à une entité de couche supérieure (voir l'IEEE 802.1D).
SendFrameReq(RPort, DestinationAddress, SourceAddress, Priority, LT, MRP-PDU)	Fonction d'envoi d'un MRP-PDU au port RPort avec SourceAddress et LT à DestinationAddress. La priorité utilisée dans TagControlInformation est codée dans la trame si le VLAN est utilisé

Nom de la fonction	Opérations
TestMgrNAckReq(MRP_Prio, MRP_SA)	Créer un MRP-PDU conformément à MRP_Option, sous-type MRP_TestMgrNAck Affectations: MRP_Type:= MRP_Option MRP_OUI:= ManufacturerOUI ou IECOU MRP_SubOption1: spécifique à un fabricant MRP_SubOption2: MRP_SubType:= SubTLVHeader.Type (= 0x01) MRP_Prio:= MRP_TS_Prio MRP_SA:= MRP_TS_SA MRP_OtherMRMPrio:= vide (0x00) MRP_OtherMRMSA:= MRP_SA du MRA à distance MRP_Type:= MRP_Common MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID SendFrameReq (RPort_1, MC_TEST, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_TEST, SA_Port2, PRIORITY, LT, MRP-PDU)
TestPropagateReq(MRP_Prio, MRP_SA)	Créer un MRP-PDU conformément à MRP_Option, sous-type MRP_TestPropagate Affectations: MRP_Type:= MRP_Option MRP_OUI:= ManufacturerOUI ou IECOU MRP_SubOption1: spécifique à un fabricant MRP_SubOption2: MRP_SubType:= SubTLVHeader.Type (= 0x02) MRP_Prio:= MRP_TS_Prio MRP_SA:= MRP_TS_SA MRP_OtherMRMPrio:= MRP_Prio du MRA à distance dont la priorité est la plus élevée MRP_OtherMRMSA:= MRP_SA du MRA à distance dont la priorité est la plus élevée MRP_Type:= MRP_Common MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID SendFrameReq (RPort_1, MC_TEST, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_TEST, SA_Port2, PRIORITY, LT, MRP-PDU)
TestMgrNAckInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA)	Recevoir un MRP-PDU conformément à MRP_Option, sous-type MRP_TestMgrNAck RPort:= R_Port MRP_SA:= MRP_SA de MRP-PDU MRP_Prio:= MRP_Prio de MRP-PDU MRP_BestMRM_SA:= MRP_SA de MRP-PDU

Nom de la fonction	Opérations
TestPropagateInd(RPort, MRP_SA, MRP_Prio, MRP_BestMRM_SA, MRP_BestMRM_Prio)	Recevoir un MRP-PDU conformément à MRP_Option, sous-type MRP_TestPropagateInd RPort:= R_Port MRP_SA:= MRP_SA de MRP-PDU MRP_Prio:= MRP_Prio de MRP-PDU MRP_BestMRM_SA:= MRP_OtherMRMSA de MRP-PDU MRP_BestMRM_Prio:= MRP_OtherMRMPrio de MRP-PDU
MRC_Init()	MRP_MonNReturn:= 0 MRP_LNKNReturn:= MRP_LNKNRmax AddMACFDB({RPort_1,RPort_2}, {MC_TEST, MC_CONTROL},ORG) MRP_BestMRM_SA:= HO_BestMRM_SA MRP_BestMRM_Prio:= HO_BestMRM_Prio
MRM_Init()	AddTest:= FALSE NO_TC:= FALSE MRP_MRM_NReturn:= 0 AddMACFDB({}, {MC_TEST, MC_CONTROL},ORG) MRP_BestMRM_SA:= MRP_TS_SA MRP_BestMRM_Prio:= MRP_TS_Prio
REM_PRIO_BETTER_THAN_OWN_PRIO	Macro pour vérifier si la priorité de l'appareil à distance est plus élevée que la sienne? (MRP_remotePrio < MRP_myPrio) ((MRP_remotePrio == MRP_myPrio) && (MRP_remoteSA < MRP_mySA)) TRUE:= l'appareil à distance est doté d'une priorité plus élevée FALSE:= l'hôte local est doté d'une priorité plus élevée
REM_PRIO_BETTER_THAN_HO_PRIO	Macro pour vérifier si la priorité de l'appareil à distance est plus élevée que la priorité la plus élevée des gestionnaires enregistrés par leur propre hôte. (MRP_remotePrio < HO_BestMRM_Prio) ((MRP_remotePrio == HO_BestMRM_Prio) && (MRP_remoteSA < HO_BestMRM_SA)) TRUE:= l'appareil à distance est doté d'une priorité plus élevée FALSE:= les gestionnaires enregistrés sont dotés d'une priorité plus élevée

8.2.5 Temporisateur d'annulation de la FDB

Le temporisateur d'annulation de la FDB est un diagramme d'états auxiliaire. Le temporisateur d'annulation de la FDB doit être conforme au diagramme d'états présenté au Tableau 47.

Tableau 47 – Temporisateur d'annulation de la FDB de filtrage du MRP

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
1	Power On	=> FDBCclearTimer.ini	IDLE
2	IDLE	FDBCclearTimer .expired => ClearLocalFDB()	IDLE

8.2.6 Temporisateur de changement de topologie

Le temporisateur de changement de topologie est un diagramme d'états auxiliaire. Il doit être conforme au diagramme d'états présenté au Tableau 48.

Tableau 48 – Temporisateur de changement de topologie du MRP

#	Etat en cours	Événement /Condition =>Action	Etat suivant
1	Power On	=> TopTimer.ini TC_NReturn:= MRP_TOPNRmax – 1 TopTimer.isRunning = FALSE	IDLE
2	IDLE	Expiration de TopTimer /TC_NReturn > 0 => SetupTopologyChangeReq(TC_NReturn X MRP_TOPchgT) TC_NReturn:= TC_NReturn – 1 TopTimer.start(MRP_TOPchgT)	IDLE
3	IDLE	Expiration de TopTimer /TC_NReturn == 0 => TC_NReturn:= MRP_TOPNRmax – 1 CLEAR_FDB(0) SetupTopologyChangeReq(0) TopTimer.isRunning = FALSE	IDLE

8.2.7 Machine de protocole du MIM

La machine de protocole du MIM en LC-mode est définie au Tableau 50 et la machine de protocole du MIM en RC-mode est définie au Tableau 51. Le comportement principal des deux machines de protocole est présenté à la Figure 14.

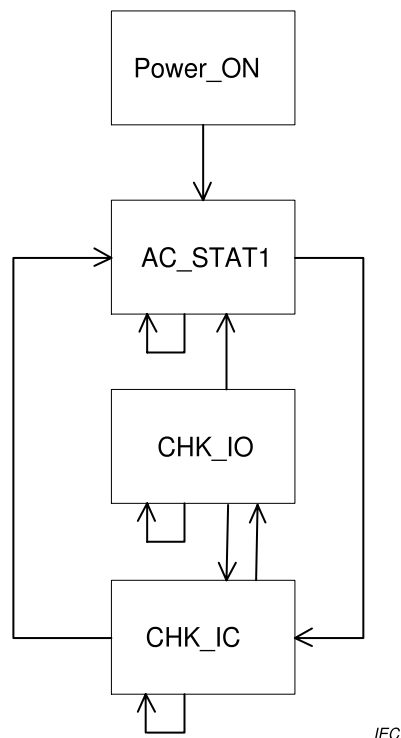


Figure 14 – Machine de protocole MRP pour le MIM en RC-mode et LC-mode

Le texte ci-dessous explique les principales actions réalisées dans les états. En cas de différence d'interprétation entre ce texte et le diagramme d'états, ce dernier l'emporte.

PowerOn

Initialisation, le MIM doit commencer avec le port d'interconnexion IPort à l'état BLOCKED. Les entrées statiques de la FDB pour les adresses multidiffusion du MRP sont générées:

- En RC-mode: MC_INTEST et MC_INCONTROL. Transfert des trames du MRP vers MC_INTEST et l'hôte.
- En LC-mode: MC_INCONTROL.

Transfert des trames du MRP vers MC_INCONTROL entre les ports en anneau et vers l'hôte. Tous les MRP-PDU doivent utiliser la priorité la plus élevée (ORG).

AC_STAT1 (état en attente de connexion 1)

Démarrage, attente de l'établissement de liaison au niveau du port d'interconnexion puis démarrage de la surveillance d'essai de l'interconnexion s'il est en RC_MODE.

CHK_IO (Check Interconnection, Interconnection Open State)

Le MIM établit une liaison au niveau de son port d'interconnexion et doit configurer MRP_InState en état d'interconnexion ouverte dans les cas suivants:

- En RC-mode: Le MIM n'a pas reçu ses trames MRP_InTest pendant une durée déterminée ou a reçu une trame MRP_InLinkDown.
- En LC-mode: Le MIM a reçu une trame MRP_InLinkDown

CHK_IC (Check Interconnection, Interconnection Closed State)

Le MIM établit une liaison au niveau de son port d'interconnexion et doit configurer MRP_InState en état d'interconnexion fermée dans les cas suivants:

- En RC-mode: Le MIM reçoit ses trames MRP_InTest
- En LC-mode: Le MIM a reçu une trame MRP_InLinkUp d'un MIC dont l'attribut Interconnection ID est égal à celui des MIM.

Les variables locales de la machine de protocole du MIM figurent au Tableau 49.

Tableau 49 – Variables locales du MRP de la machine de protocole du MIM

Nom	Type	Signification
RPort_1	Unsigned16	Identification de port du port en anneau 1
RPort_2	Unsigned16	Identification de port du port en anneau 2
IPort	Unsigned16	Identification de port du port d'interconnexion
SA_Port1	OctetString[6]	Adresse source MAC RPort_1 du port en anneau
SA_Port2	OctetString[6]	Adresse source MAC RPort_2 du port en anneau
SA_IPort	OctetString[6]	Adresse source MAC IPort du port d'interconnexion
PRIORITY	Unsigned8	Priorité conforme à l'IEEE 802.1Q pour les MRP-PDU. Doit être définie sur ORG.
MRP_TS_SA	OctetString[6]	Adresse source MAC de l'hôte
MRP_MIM_NRmax	Unsigned16	Nombre maximal de retransmissions de MRP-PDU de type MRP_InTest
MRP_MIM_NReturn	Unsigned16	Compteur, plage MRP_MIM_NRmax sur 0
IN_TC_NReturn	Unsigned16	Compteur, plage MRP_IN_TOPNRmax sur 0
MRP_IN_LNKSTAT_NReturn	Unsigned16	Compteur, plage MRP_IN_LNKSTATNRmax sur 0
IID	Unsigned16	Identification du domaine d'interconnexion
MRP_LNK_UP	Unsigned16	Valeur constante pour indiquer un établissement de liaison
MRP_LNK_DOWN	Unsigned16	Valeur constante pour indiquer une interruption de liaison

Le diagramme d'états du MIM en LC-mode doit être conforme au Tableau 50.

Tableau 50 – Diagramme d'états du MIM en LC-mode

#	Etat en cours	Événement /Condition =>Action	Etat suivant
1	PowerOn	=> InitFDB() AddMACFDB({RPort_1,RPort_2}, {MC_INCONTROL},ORG) AddMACFDB({local},{MC_INCONTROL}, ORG) SetPortStateReq (IPort, BLOCKED) IID:= identifiant d'interconnexion	AC_STAT1
2	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => SetPortStateReq(IPort, BLOCKED) InterconnLinkStatusPollReq (MRP_IN_LNKSTATchgT)	CHK_IC
3	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => Ignore	AC_STAT1
4	AC_STAT1	InterconnLinkChangeInd(InID, Link_status, RPort) => Ignore	AC_STAT1
5	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	AC_STAT1
6	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	AC_STAT1
7	CHK_IO	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => Ignore	CHK_IO
8	CHK_IO	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, BLOCKED) InterconnLinkStatusTimer.stop	AC_STAT1

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
9	CHK_IO	InterconnLinkChangeInd (InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_UP => SetPortStateReq (IPort, BLOCKED) InterconnLinkStatusTimer.stop InterconnTopologyChangeReq (MRP_IN_TOPchgT)	CHK_IC
10	CHK_IO	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_DOWN => SetPortStateReq (IPort, FORWARDING)	CHK_IO
11	CHK_IO	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	CHK_IO
12	CHK_IO	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	CHK_IO
13	CHK_IC	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => Ignore	CHK_IC
14	CHK_IC	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, BLOCKED)	AC_STAT1
15	CHK_IC	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, FORWARDING) InterconnLinkStatusTimer.stop InterconnTopologyChangeReq(MRP_IN_TOPchgT)	CHK_IO
16	CHK_IC	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_UP => Ignore	CHK_IC

#	Etat en cours	Événement /Condition =>Action	Etat suivant
17	CHK_IC	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	CHK_IC
18	CHK_IC	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	CHK_IC

Le diagramme d'états du MIM en RC-mode doit être conforme au Tableau 51.

Tableau 51 – Diagramme d'états du MIM en RC-mode

#	Etat en cours	Événement /Condition =>Action	Etat suivant
1	PowerOn	=> InitFDB() AddMACFDB({RPort_1,RPort_2}, {MC_INCONTROL},ORG) AddMACFDB({local},{MC_INCONTROL, MC_INTEST}, ORG) SetPortStateReq (IPort, BLOCKED)InterconnTestTimer.ini IID:= identifiant d'interconnexion	AC_STAT1
2	AC_STAT1	Expiration d'InterconnTestTimer => Ignore	AC_STAT1
3	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP >= SetPortStateReq(IPort, BLOCKED) MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0 InterconnTestReq(MRP_IN_TSTdefaultT)	CHK_IC
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => Ignore	AC_STAT1
5	AC_STAT1	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA == MRP_TS_SA => SetPortStateReq(IPort, BLOCKED) MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0 InterconnTestReq(MRP_IN_TSTdefaultT)	CHK_IC
6	AC_STAT1	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_1 => InterconnForwardReq(RPort_2)	AC_STAT1
7	AC_STAT1	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_2 => InterconnForwardReq(RPort_1)	AC_STAT1
8	AC_STAT1	InterconnLinkChangeInd(InID, Link_status, RPort) => Ignore	AC_STAT1

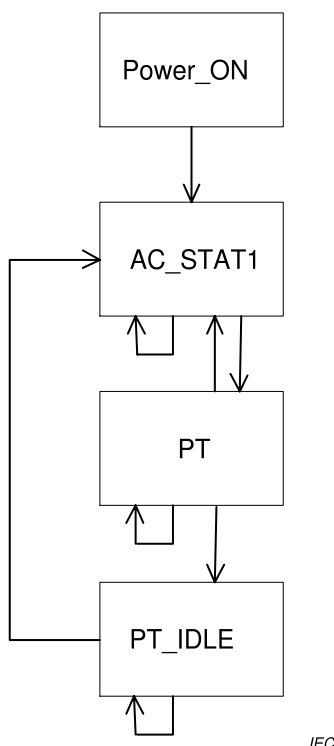
#	Etat en cours	Evénement /Condition =>Action	Etat suivant
9	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	AC_STAT1
10	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	AC_STAT1
11	CHK_IO	Expiration d'InterconnTestTimer => InterconnTestReq(MRP_IN_TSTdefaultT)	CHK_IO
12	CHK_IO	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => Ignore	CHK_IO
13	CHK_IO	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, BLOCKED) InterconnTopologyChangeReq(MRP_IN_TOPchgT) InterconnTestReq(MRP_IN_TSTdefaultT)	AC_STAT1
14	CHK_IO	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA == MRP_TS_SA => SetPortStateReq (IPort, BLOCKED) MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0 InterconnTopologyChangeReq (MRP_IN_TOPchgT) InterconnTestReq (MRP_IN_TSTdefaultT)	CHK_IC
15	CHK_IO	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_1 => InterconnForwardReq(RPort_2)	CHK_IO
16	CHK_IO	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_2 => InterconnForwardReq(RPort_1)	CHK_IO

#	Etat en cours	Événement /Condition =>Action	Etat suivant
17	CHK_IO	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_UP => InterconnTestReq(MRP_IN_TSTdefaultT)	CHK_IO
18	CHK_IO	InterconnLinkChangeInd(InID, Link_status, RPort) /Link_status == MRP_LNK_DOWN => Ignore	CHK_IO
19	CHK_IO	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	CHK_IO
20	CHK_IO	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	CHK_IO
21	CHK_IC	InterconnTestTimer expired /MRP_MIM_NReturn >= MRP_MIM_NRmax => SetPortStateReq (IPort, FORWARDING) MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0 InterconnTopologyChangeReq (MRP_IN_TOPchgT) InterconnTestReq (MRP_IN_TSTdefaultT)	CHK_IO
22	CHK_IC	InterconnTestTimer expired /MRP_MIM_NReturn < MRP_MIM_NRmax => MRP_MIM_NReturn:= MRP_MIM_NReturn + 1 InterconnTestReq (MRP_IN_TSTdefaultT)	CHK_IC
23	CHK_IC	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => Ignore	CHK_IC
24	CHK_IC	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, BLOCKED) InterconnTopologyChangeReq(MRP_IN_TOPchgT) InterconnTestReq(MRP_IN_TSTdefaultT)	AC_STAT1

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
25	CHK_IC	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA == MRP_TS_SA => MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 MRP_MIM_NReturn:= 0	CHK_IC
26	CHK_IC	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_1 => InterconnForwardReq(RPort_2)	CHK_IC
27	CHK_IC	InterconnTestInd (MRP_SA, RPort, InID) /MRP_SA != MRP_TS_SA && RPort == RPort_2 => InterconnForwardReq(RPort_1)	CHK_IC
28	CHK_IC	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_DOWN => SetPortStateReq(IPort, FORWARDING) InterconnTopologyChangeReq(MRP_IN_TOPchgT)	CHK_IO
29	CHK_IC	InterconnLinkChangeInd(InID, Link_status, RPort) /InID == IID && Link_status == MRP_LNK_UP => MRP_MIM_NRmax:= MRP_IN_TSTNRmax – 1 InterconnTopologyChangeReq (MRP_IN_TOPchgT)	CHK_IC
30	CHK_IC	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA != MRP_TS_SA => ClearFDB(t)	CHK_IC
31	CHK_IC	InterconnTopologyChangeInd(MRP_SA, t, InID) /MRP_SA == MRP_TS_SA => Ignore	CHK_IC

8.2.8 Machine de protocole du MIC

La machine de protocole du MIC en LC-mode est définie au Tableau 53 et la machine de protocole du MIC en RC-mode est définie au Tableau 54. Le comportement principal de la machine de protocole est présenté à la Figure 15.



IEC

Figure 15 – Machine de protocole MRP pour le MIC en RC-mode et LC-mode

Le texte ci-dessous explique les principales actions réalisées dans les états. En cas de différence d'interprétation entre ce texte et le diagramme d'états, ce dernier l'emporte.

PowerOn

Initialisation, le MIC doit commencer avec le port d'interconnexion IPort à l'état BLOCKED. Les entrées statiques de la FDB pour les adresses multidiffusion du MRP sont générées:

- En RC-mode: MC_INTEST et MC_INCONTROL. Transfert des trames du MRP vers MC_INTEST entre les ports en anneau et vers le port d'interconnexion.
- En LC-mode: MC_INCONTROL.

Transfert des trames du MRP vers MC_INCONTROL dans les deux modes entre les ports en anneau et vers l'hôte. Tous les MRP-PDU doivent utiliser la priorité la plus élevée (ORG).

AC_STAT1 (état en attente de connexion 1)

Démarrage, attente de l'établissement de liaison au niveau du port d'interconnexion.

PT (transmission)

Etat temporaire lors du signalement des changements de liaison.

IP_IDLE (état en attente du port d'interconnexion)

Cet état doit être atteint si le port d'interconnexion comporte une liaison et que l'état de son port est FORWARDING.

Les variables locales de la machine de protocole du MIC figurent au Tableau 52.

Tableau 52 – Variables locales du MRP de la machine de protocole du MIC

Nom	Type	Signification
RPort_1	Unsigned16	Identification de port du port en anneau 1
RPort_2	Unsigned16	Identification de port du port en anneau 2
IPort	Unsigned16	Identification de port du port d'interconnexion
SA_Port1	OctetString[6]	Adresse source MAC RPort_1 du port en anneau
SA_Port2	OctetString[6]	Adresse source MAC RPort_2 du port en anneau
SA_IPort	OctetString[6]	Adresse source MAC IPort du port d'interconnexion
PRIORITY	Unsigned8	Priorité conforme à l'IEEE 802.1Q pour les MRP-PDU. Doit être définie sur ORG.
MRP_TS_SA	OctetString[6]	Adresse source MAC de l'hôte
IID	Unsigned16	Identification du domaine d'interconnexion
MRP_InLNKNReturn	Unsigned16	Compteur, plage MRP_IN_LNKNRmax sur 0
MRP_LNK_UP	Unsigned16	Valeur constante pour indiquer un établissement de liaison
MRP_LNK_DOWN	Unsigned16	Valeur constante pour indiquer une interruption de liaison

Le diagramme d'états du MIC en LC-mode doit être conforme au Tableau 53.

Tableau 53 – Diagramme d'états du MIC en LC-mode

#	Etat en cours	Événement /Condition =>Action	Etat suivant
1	PowerOn	=> InitFDB() AddMACFDB({RPort_1,RPort_2}, {MC_INCONTROL},ORG) AddMACFDB({local},{MC_INCONTROL},ORG) SetPortStateReq (IPort, BLOCKED) InterconnUpTimer.ini InterconnDownTimer.ini IID:= identifiant d'interconnexion	AC_STAT1
2	AC_STAT1	Expiration d'InterconnDownTimer /MPR_InLNKNReturn == 0 => MPR_InLNKNReturn:= MPR_IN_LNKNRmax	AC_STAT1
3	AC_STAT1	Expiration d'InterconnDownTimer /MPR_InLNKNReturn 0 => MPR_InLNKNReturn:= MPR_InLNKNReturn – 1 InterconnDownTimer.start(MPR_IN_LNKdownT) InterconnLinkChangeReq (MPR_LNK_DOWN, MPR_InLNKNReturn X MPR_IN_LNKdownT)	AC_STAT1
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MPR_LNK_UP => MPR_InLNKNReturn:= MPR_IN_LNKNRmax InterconnDownTimer.stop InterconnUpTimer.start(MPR_IN_LNKupT) InterconnLinkChangeReq (MPR_LNK_UP, MPR_InLNKNReturn X MPR_IN_LNKupT)	PT
5	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /Link_status == MPR_LNK_DOWN => Ignore	AC_STAT1
6	AC_STAT1	InterconnTopologyChangeInd(MPR_SA, t, InID) /InID == IID => InterconnDownTimer.stop	AC_STAT1
7	AC_STAT1	InterconnLinkStatusInd (InID, RPort) /InID == IID => InterconnLinkChangeReq (MPR_LNK_DOWN, 0)	AC_STAT1

#	Etat en cours	Événement /Condition =>Action	Etat suivant
8	PT	Expiration d'InterconnUpTimer /MRP_InLNKNReturn == 0 => MRP_InLNKNReturn:= MRP_IN_LNKNRmax SetPortStateReq (IPort, FORWARDING)	IP_IDLE
9	PT	InterconnUpTimer expired /MRP_InLNKNReturn > 0 => MRP_InLNKNReturn:= MRP_InLNKNReturn – 1 InterconnUpTimer.start(MRP_LNKupT) InterconnLinkChangeReq (MRP_LNK_UP, MRP_InLNKNReturn X MRP_IN_LNKupT)	PT
10	PT	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnUpTimer.stop SetPortStateReq (IPort, BLOCKED) InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1
11	PT	MAUTypeChangeInd (RPort, Link_status) /Link_status == MRP_LNK_UP => Ignore	PT
12	PT	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnUpTimer.stop SetPortStateReq (IPort, FORWARDING)	IP_IDLE
13	PT	InterconnLinkStatusInd (InID, RPort) /InID == IID => InterconnLinkChangeReq (MRP_LNK_UP, 0)	PT
14	IP_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => MRP_InLNKNReturn:= MRP_IN_LNKNRmax SetPortStateReq (IPort, BLOCKED) InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
15	IP_IDLE	MAUTypeChangeInd (RPort, Link_status) && Link_status == MRP_LNK_UP => Ignore	IP_IDLE
16	IP_IDLE	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => Ignore	IP_IDLE
17	IP_IDLE	InterconnLinkStatusInd (InID, RPort) /InID == IID => InterconnLinkChangeReq (MRP_LNK_UP, 0) InterconnForwardReq(IPort)	IP_IDLE
18	IP_IDLE	InterconnLinkChangeInd(InID, LinkStatus, RPort) /RPort != IPort && InID == IID => InterconnForwardReq(IPort)	IP_IDLE

Le diagramme d'états du MIC en RC-mode doit être conforme au Tableau 54.

Tableau 54 – Diagramme d'états du MIC en RC-mode

#	Etat en cours	Événement /Condition =>Action	Etat suivant
1	PowerOn	=> InitFDB() AddMACFDB({RPort_1,RPort_2,IPort}, {MC_INTEST},ORG) AddMACFDB({RPort_1,RPort_2}, {MC_INCONTROL},ORG) AddMACFDB({local},{MC_INCONTROL},ORG) SetPortStateReq (IPort, BLOCKED) InterconnUpTimer.ini InterconnDownTimer.ini IID:= identifiant d'interconnexion	AC_STAT1
2	AC_STAT1	Expiration d'InterconnDownTimer /MRP_InLNKNReturn == 0 => MRP_InLNKNReturn:= MRP_IN_LNKNRmax	AC_STAT1
3	AC_STAT1	Expiration d'InterconnDownTimer /MRP_InLNKNReturn 0 => MRP_InLNKNReturn:= MRP_InLNKNReturn – 1 InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1
4	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_UP => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnDownTimer.stop InterconnUpTimer.start(MRP_IN_LNKupT) InterconnLinkChangeReq (MRP_LNK_UP, MRP_InLNKNReturn X MRP_IN_LNKupT)	PT
5	AC_STAT1	MAUTypeChangeInd (RPort, Link_status) /Link_status == MRP_LNK_DOWN => Ignore	AC_STAT1
6	AC_STAT1	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => InterconnDownTimer.stop	AC_STAT1
7	PT	Expiration d'InterconnUpTimer /MRP_InLNKNReturn == 0 => MRP_InLNKNReturn:= MRP_IN_LNKNRmax SetPortStateReq (IPort, FORWARDING)	IP_IDLE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
8	PT	Expiration d'InterconnUpTimer /MRP_InLNKNReturn 0 => MRP_InLNKNReturn:= MRP_InLNKNReturn 1 InterconnUpTimer.start(MRP_LNKupT) InterconnLinkChangeReq (MRP_LNK_UP, MRP_InLNKNReturn X MRP_IN_LNKupT)	PT
9	PT	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnUpTimer.stop SetPortStateReq (IPort, BLOCKED) InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1
10	PT	MAUTypeChangeInd (RPort, Link_status) /Link_status == MRP_LNK_UP => Ignore	PT
11	PT	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => MRP_InLNKNReturn:= MRP_IN_LNKNRmax InterconnUpTimer.stop SetPortStateReq (IPort, FORWARDING)	IP_IDLE
12	IP_IDLE	MAUTypeChangeInd (RPort, Link_status) /RPort == IPort && Link_status == MRP_LNK_DOWN => MRP_InLNKNReturn:= MRP_IN_LNKNRmax SetPortStateReq (IPort, BLOCKED) InterconnDownTimer.start(MRP_IN_LNKdownT) InterconnLinkChangeReq (MRP_LNK_DOWN, MRP_InLNKNReturn X MRP_IN_LNKdownT)	AC_STAT1
13	IP_IDLE	MAUTypeChangeInd (RPort, Link_status) && Link_status == MRP_LNK_UP => Ignore	IP_IDLE
14	IP_IDLE	InterconnTopologyChangeInd(MRP_SA, t, InID) /InID == IID => Ignore	IP_IDLE

#	Etat en cours	Événement /Condition =>Action	Etat suivant
15	IP_IDLE	InterconnLinkChangeInd(InID, LinkStatus, RPort) /RPort != IPort && InID == IID => InterconnForwardReq(IPort)	IP_IDLE

8.2.9 Fonctions du MIM et du MIC

Les fonctions du MIM et du MIC doivent être conformes au Tableau 55.

Tableau 55 – Fonctions d'interconnexion du MRP

Nom de la fonction	Opérations
InterconnTestReq (time)	SetupInterconnTestReq() InterconnTestTimer.start(time)
SetupInterconnTestReq()	Créer un MRP-PDU conformément à MRP_InTest Affectations: MRP_Type:= MRP_InTest MRP_InID:= IID MRP_SA:= MRP_TS_SA MRP_PortRole:= trame envoyée sur le port en anneau principal ou secondaire ou sur le port d'interconnexion MRP_InState:= état en cours d'interconnexion MRP_Transition:= nombre réel de transitions entre les états d'interconnexion ouverte et fermée MRP_TimeStamp:= valeur en cours du compteur local MRP_Type:= MRP_Common MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (IPort, MC_INTEST, SA_IPort, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_1, MC_INTEST, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_INTEST, SA_Port2, PRIORITY, LT, MRP-PDU)
InterconnTestInd (MRP_SA, RPort, InID)	Recevoir un MRP-PDU conformément à MRP_InTest MRP_SA:= MRP_SA de MRP-PDU RPort:= port qui a reçu ce paquet InID:= MRP_InID de MRP-PDU
InterconnForwardReq(RPort)	Fonction pour envoyer le MRP-PDU reçu sur le port RPort.
InterconnTopologyChangeReq(time)	SetupInterconnTopologyChangeReq(MRP_IN_TOPNRmax X time) si time == 0 ClearLocalFDB() sinon InterconnTopTimer.start(MRP_IN_TOPchgT)

Nom de la fonction	Opérations
SetupInterconnTopologyChangeReq(time)	Créer un MRP-PDU conformément à MRP_InTopologyChange Affectations: MRP_Type:= MRP_InTopologyChange MRP_InID:= IID MRP_SA:= MRP_TS_SA MRP_Interval:= time MRP_Type:= MRP_Common MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_INCONTROL, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_INCONTROL, SA_Port2, PRIORITY, LT, MRP-PDU) SendFrameReq (IPort, MC_INCONTROL, SA_IPort, PRIORITY, LT, MRP-PDU)
InterconnTopologyChangeInd(MRP_SA, time, InID)	Recevoir un MRP-PDU conformément à MRP_InTopologyChange MRP_SA:= MRP_SA de MRP-PDU time:= MRP_Interval de MRP-PDU InID:= MRP_InID de MRP-PDU
InterconnLinkChangeReq(LinkStatus, time)	Créer un MRP-PDU conformément à MRP_InLinkUp ou MRP_InLinkDown Affectations: si LinkStatus == MRP_LNK_UP MRP_Type:= MRP_LinkUp sinon MRP_Type:= MRP_LinkDown MRP_InID:= IID MRP_SA:= MRP_TS_SA MRP_PortRole:=: trame envoyée sur le port en anneau principal ou secondaire ou sur le port d'interconnexion MRP_Interval:= time MRP_Type:= MRP_Common MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_INCONTROL, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_INCONTROL, SA_Port2, PRIORITY, LT, MRP-PDU) SendFrameReq (IPort, MC_INCONTROL, SA_IPort, PRIORITY, LT, MRP-PDU)

Nom de la fonction	Opérations
InterconnLinkChangeInd(InID, LinkStatus, RPort)	Recevoir un MRP-PDU conformément à MRP_InLinkDown ou MRP_InLinkUp InID:= MRP_InID de MRP-PDU RPort:= port qui a reçu ce paquet si MRP_Type == MRP_InLinkUp LinkStatus:= MRP_LNK_UP sinon LinkStatus:= MRP_LNK_DOWN
InterconnLinkStatusPollReq(time)	SetupInterconnLinkStatusPollReq() Si time = 0 InterconnLinkStatusTimer.start(MRP_IN_LNKSTATchgT)
SetupInterconnLinkStatusPollReq()	Créer un MRP-PDU conformément à MRP_In_LinkStatusPollRequest Affectations: InID:= IID MRP_SA:= MRP_TS_SA MRP_PortRole:= trame envoyée sur le port en anneau principal ou secondaire ou sur le port d'interconnexion MRP_Type:= MRP_Common MRP_SequenceID:= SequenceID suivant MRP_DomainUUID:= domainUUID MRP_Type:= MRP_End SendFrameReq (RPort_1, MC_INCONTROL, SA_Port1, PRIORITY, LT, MRP-PDU) SendFrameReq (RPort_2, MC_INCONTROL, SA_Port2, PRIORITY, LT, MRP-PDU)
InterconnLinkStatusInd(InID, RPort)	Recevoir un MRP-PDU conformément à MRP_InLinkStatusPoll InID:= MRP_InID de MRP-PDU RPort:= port qui a reçu ce paquet
MAUTypeChangeInd(RPort, Link_status)	Recevoir une indication de changement de liaison locale. RPort:= port à l'origine de l'indication de changement de liaison locale. Link_status:= MRP_LNK_UP ou MRP_LNK_DOWN (en fonction de l'indication de changement de liaison locale)
SetPortStateReq(RPort, Status)	Fonction qui permet de configurer l'état de port de RPort sur Status
ClearFDB(time)	FDBClearTimer.start(time)
ClearLocalFDB()	Fonction qui permet d'annuler la FDB à l'intérieur du nœud du MRP. L'apprentissage des adresses sources à partir des trames entrantes, qui sont envoyées avant le signalement du changement de topologie, doit être évité
InitFDB()	Fonction d'initialisation de la base de données de filtrage
AddMACFDB(Destination, MAC-Address, Priority)	Fonction d'ajout d'entrées de filtrage statiques (adresse MAC) dans la FDB avec la priorité et la destination. Dans le schéma d'état, le terme "local" signifie une connexion à une entité de couche supérieure (voir l'IEEE 802.1D).

Nom de la fonction	Opérations
SendFrameReq(RPort, DestinationAddress, SourceAddress, Priority, LT, MRP-PDU)	Fonction d'envoi d'un MRP-PDU au port RPort avec SourceAddress et LT à DestinationAddress. La priorité utilisée dans TagControlInformation est codée dans la trame si le VLAN est utilisé

8.2.10 Temporisateur de changement de topologie d'interconnexion

Le temporisateur de changement de topologie d'interconnexion est un diagramme d'états auxiliaire. Le temporisateur de changement de topologie d'interconnexion doit être conforme au diagramme d'états présenté au Tableau 56.

Tableau 56 – Temporisateur de changement de topologie d'interconnexion du MRP

#	Etat en cours	Evénement /Condition =>Action	Etat suivant
1	PowerOn	=> InterconnTopTimer.ini IN_TC_NReturn:= MRP_IN_TOPNRmax – 1	IDLE
2	IDLE	Expiration d'InterconnTopTimer /IN_TC_NReturn 0 => SetupInterconnTopologyChangeReq (IN_TC_NReturn X MRP_IN_TOPchgT) IN_TC_NReturn:= IN_TC_NReturn – 1 InterconnTopTimer.start(MRP_IN_TOPchgT)	IDLE
3	IDLE	Expiration d'InterconnTopTimer /IN_TC_NReturn == 0 => IN_TC_NReturn:= MRP_IN_TOPNRmax – 1 ClearFDB(0) SetupInterconnTopologyChangeReq (0)	IDLE

8.2.11 Temporisateur d'interrogation sur l'état de liaison d'interconnexion

Le temporisateur d'interrogation sur l'état de liaison d'interconnexion est un diagramme d'états auxiliaire. Il doit être conforme au diagramme d'états présenté au Tableau 57.

Tableau 57 – Temporisateur d'interrogation sur l'état de liaison d'interconnexion du MRP

#	Etat en cours	Événement /Condition =>Action	Etat suivant
1	PowerOn	=> InterconnLinkStatusTimer.ini MRP_IN_LNKSTAT_NReturn:= MRP_IN_LNKSTATNRmax – 1	IDLE
2	IDLE	Expiration d'InterconnLinkStatusTimer /MRP_IN_LNKSTAT_NReturn 0 => SetupInterconnLinkStatusPollReq () MRP_IN_LNKSTAT_NReturn:= MRP_IN_LNKSTAT_NReturn – 1 InterconnLinkStatusTimer.start(MRP_IN_LNKSTATchgT)	IDLE
3	IDLE	Expiration d'InterconnLinkStatusTimer /MRP_IN_LNKSTAT_NReturn == 0 => RP_IN_LNKSTAT_NReturn:= MRP_IN_LNKSTATNRmax – 1 SetupInterconnLinkStatusPollReq ()	IDLE

9 Installation, configuration et réparation de MRP

9.1 Paramètres du port d'interconnexion et du port en anneau

La configuration du port en anneau pour le MRM et tous les MRC d'un anneau doit satisfaire aux paramètres figurant au Tableau 58.

La configuration du port d'interconnexion du MRP pour le MIM et tous les MIC d'une interconnexion doit satisfaire aux paramètres figurant au Tableau 58.

Tableau 58 – Paramètres de réseau/connexion du MRP

Paramètre	Valeur
Link speed	La vitesse de liaison minimale doit être de 100 Mbit/s
Duplex setting	Les ports en anneau et le port d'interconnexion doivent fonctionner en mode duplex. Le mode administratif d'un port peut être défini sur auto-négociation, mais la valeur négociée (mode opér) doit être en mode duplex

9.2 Paramètres de topologie en anneau

Le nombre de nœuds actifs dans un anneau ne doit pas dépasser 50.

NOTE 1 Si un anneau contient plus que ce nombre de nœuds, le temps de reprise maximal peut être dépassé et l'anneau peut devenir instable.

NOTE 2 Si le temps de reprise maximal est de 10 ms, le nombre de nœuds peut être inférieur au nombre indiqué ci-dessus (voir 9.9 pour plus de détails).

9.3 Paramètres du MRM

En configurant ses paramètres, le MRM définit le temps de reprise maximal d'un anneau. Le Tableau 59 spécifie quatre ensembles cohérents de paramètres correspondant à un temps de reprise de l'anneau de 500 ms, 200 ms, 30 ms et 10 ms.

NOTE MRM et MRC peuvent prendre en charge des ensembles de paramètres cohérents supplémentaires correspondant à des temps de reprise plus courts ou plus longs. Le concepteur est chargé de la cohérence de tous les paramètres présentés à l'Article 9 (voir les exemples en 9.9). L'installateur est chargé de la cohérence de tous les nœuds de l'anneau.

Tableau 59 – Paramètres relatifs au MRM du MRP

Paramètre	Temps de reprise maximal				Signification
	500 ms	200 ms	30 ms ^b	10 ms ^b	
MRP_TOPchgT	20 ms	10 ms	0,5 ms	0,5 ms	Intervalle de demande de changement de topologie (table d'annulation d'adresse)
MRP_TOPNRmax	3	3	3	3	Nombre de répétitions de changement de topologie (table d'annulation d'adresse)
MRP_TSTshortT	30 ms	10 ms	1 ms ^c	0,5 ms ^c	Intervalle court MRP_Test
MRP_TSTdefaultT	50 ms	20 ms	3,5 ms ^c	1 ms ^c	Intervalle par défaut MRP_Test
MRP_TSTNRmax	5	3	3	3	Nombre de surveillances MRP_Test
MRP_TSTExtNRmax	15	N.A ^a	N.A ^a	N.A ^a	Nombre de surveillances étendues MRP_Test (option)
^a Il est exigé que les instructions suivantes soient respectées: La valeur FALSE doit être affectée à l'option "MRC non bloquant pris en charge" pour un temps de reprise maximal de 200 ms, 30 ms et 10 ms. Tous les MRC doivent prendre en charge le mode de blocage. MRP_TSTExtNRmax n'est pas applicable. ^b Les temps de reprise 10 ms et 30 ms impliquent des restrictions supplémentaires quant à la longueur du câble et au temps de réaction de chaque client dans le réseau (voir 9.4 et 9.9 pour plus de détails). ^c Pour limiter la charge sur les appareils du réseau en cas d'erreur RING_OPEN (du fait d'une mauvaise configuration, par exemple), les intervalles de MRP_TSTshortT et MRP_TSTdefaultT peuvent être configurés de manière à prendre des valeurs plus élevées dans ce cas.					

9.4 Paramètres et contraintes pour le MRC

Le Tableau 60 spécifie les ensembles de paramètres du MRC (utilisés pour les quatre ensembles de paramètres du MRM).

Tableau 60 – Paramètres relatifs au MRC du MRP

Paramètre	Temps de reprise max. 500 ms et 200 ms	Temps de reprise max. 30 ms et 10 ms	Signification
MRP_LNKdownT	20 ms	1 ms	Intervalle du temporisateur d'interruption de liaison
MRP_LNKupT	20 ms	1 ms	Intervalle du temporisateur d'établissement de liaison
MRP_LNKNRmax	4	4	Nombre de changements de liaison (établissement ou interruption)

NOTE 1 Ces paramètres sont calculés selon l'hypothèse que la charge de trafic dans l'anneau ne dépasse pas 90 %.

Pour les temps de reprise 10 ms et 30 ms, le MRC doit garantir:

- un vidage rapide de sa FDB; voir 9.9 pour plus de détails;
- un changement d'état du port (de BLOCKED à NONBLOCKED, par exemple): < 0,5 ms
- une conception pour un temps de réaction court d'un commutateur par rapport à un événement de prise en charge d'une rapide reconfiguration.

NOTE 2 Certains de ces paramètres peuvent être mesurés conformément à l'IEC 62439-1:2010 et l'IEC 62439-1:2010/AMD1:2012, Article 8.

9.5 Compatibilité du MRA avec des versions antérieures du protocole du gestionnaire automatique

En plus de la fonctionnalité du MRA décrite dans le présent document, l'utilisateur d'un appareil compatible MRA peut demander la prise en charge des versions antérieures du protocole du gestionnaire automatique à des fins de rétrocompatibilité.

Le trajet de migration vers la fonction normale du gestionnaire automatique peut être:

- la topologie en anneau n'utilise pas les fonctionnalités du gestionnaire automatique et tous les appareils doivent être configurés manuellement pour le rôle MRM ou MRC. Un seul appareil doit être configuré pour le rôle MRM; les autres appareils doivent être configurés pour le rôle MRC;
- la topologie en anneau utilise la fonctionnalité du gestionnaire automatique décrite dans la présente norme. Les appareils hérités qui ne prennent pas le gestionnaire automatique en charge comme défini dans la présente norme doivent être configurés manuellement pour le rôle MRC. Les appareils qui prennent en charge la fonctionnalité de gestionnaire automatique en charge comme défini dans la présente norme peuvent participer au processus de vote et voter pour le MRM;
- la topologie en anneau utilise le protocole de gestionnaire automatique plus ancien qui n'est pas totalement conforme avec la fonctionnalité de gestionnaire automatique décrite dans la présente norme. Il est dans ce cas nécessaire, à des fins de migration, que les appareils conformes à la présente norme prennent en charge la gestion des champs MRP_Option et MRP_Suboption supplémentaires, comme décrit à l'Annexe A. Cette solution constitue un chemin de migration valide vers la fonction de gestionnaire automatique introduite dans le présent document; elle protège ainsi la base installée d'appareils de terrain.

9.6 Paramètres de topologie d'interconnexion

Les liaisons d'interconnexion entre deux anneaux du MRP ne doivent contenir aucun nœud supplémentaire.

NOTE Dans une liaison d'interconnexion, les anneaux connectés deviennent instables si un MRC détecte une défaillance sur un port en anneau et signale ce changement en envoyant une trame MRP_LinkDown à l'anneau.

9.7 Paramètres du MIM

En configurant son jeu de paramètres, le MIM définit le temps de reprise maximal d'une interconnexion du MRP. Le Tableau 61 spécifie deux ensembles cohérents de paramètres correspondant à des temps de reprise de l'interconnexion du MRP de 500 ms et 200 ms.

NOTE MIM et MIC peuvent prendre en charge des ensembles de paramètres cohérents supplémentaires correspondant à des temps de reprise plus courts ou plus longs. Le concepteur est chargé de la cohérence de tous les paramètres présentés à l'Article 9 (voir les exemples en 9.9). L'installateur est chargé de la cohérence de tous les nœuds de l'anneau et de son interconnexion.

Tableau 61 – Paramètres relatifs au MIM du MRP

Paramètre	Temps de reprise maximal		Signification
	500 ms	200 ms	
MRP_IN_TOPchgT	20 ms	10 ms	Intervalle de demande de changement de topologie d'interconnexion (table d'annulation d'adresse)
MRP_IN_TOPNRmax	3	3	Nombre de répétitions de changement de topologie d'interconnexion (table d'annulation d'adresse)
MRP_IN_TSTdefaultT	50 ms ^a	20 ms ^a	Intervalle par défaut MRP_InTest
MRP_IN_TSTNRmax	8	8	Nombre de surveillances MRP_InTest
MRP_IN_LNKSTATchgT	20 ms	20 ms	Intervalle d'interrogation de statut de liaison d'interconnexion
MRP_IN_LNKSTATNRmax	8	8	Nombre de surveillances MRP_InLinkStatusPoll
^a Pour limiter la charge sur les appareils du réseau en cas de valeur INTERCONNECTION_OPEN (du fait d'une mauvaise configuration, par exemple), l'intervalle de MRP_IN_TSTdefaultT peut être configuré de manière à prendre des valeurs plus élevées dans ce cas.			

9.8 Paramètres et contraintes pour le MIC

Le Tableau 62 spécifie les ensembles de paramètres du MIC (utilisés pour tous les ensembles de paramètres du MIM).

Tableau 62 – Paramètres relatifs au MIC du MRP

Paramètre	Temps de reprise max. 500 ms et 200 ms	Signification
MRP_IN_LNKdownT	20 ms	Intervalle du temporisateur d'interruption de liaison d'interconnexion
MRP_IN_LNKupT	20 ms	Intervalle du temporisateur d'établissement de liaison d'interconnexion
MRP_IN_LNKNRmax	4	Nombre de changements de liaison d'interconnexion (établissement ou interruption)
NOTE Ces paramètres sont calculés en s'appuyant sur l'hypothèse que la charge de trafic dans l'anneau et dans l'interconnexion de l'anneau ne dépasse pas 90 %.		

NOTE Certains de ces paramètres peuvent être mesurés conformément à l'IEC 62439-1:2010 et l'IEC 62439-1:2010/AMD1:2012, Article 8.

9.9 Calcul du temps de reprise de l'anneau du MRP

9.9.1 Vue d'ensemble

Pour un temps de reprise de 10 ms, le temps de reprise réel du réseau dépend également du nombre d'appareils participant. Le temps de transfert de chaque appareil et des trames de délai lors du déplacement dans le réseau en anneau doit être pris en considération. Pour calculer le temps réel de reconfiguration d'un réseau, les formules ci-après décrites en 9.9.2 peuvent être utilisées. En 9.9.3, un exemple de calcul est donné pour le scénario le plus défavorable. En 9.9.4, un exemple permet d'obtenir le meilleur temps de reconfiguration possible pour 50 appareils avec l'ensemble de paramètres cohérents pour 30 ms.

NOTE Compte tenu de la fréquence élevée des trames d'essai pour les temps de reprise 10 ms et 30 ms, la probabilité de détection d'une erreur via les trames d'essai est plus importante que la probabilité de détection via des messages d'interruption de liaison. Par conséquent, les calculs suivants ne concernent pas l'éventuelle détection plus rapide par des messages d'interruption de liaison.

9.9.2 Dédution de formule

Le temps de reprise réel de l'anneau dépend de différentes variables, qui doivent toutes être factorisées de la manière décrite ci-dessous:

Un MRM détecte une défaillance de l'anneau, alors qu'il n'a pas reçu ses trames d'essai au bout d'une durée définie. Cette durée T_{test} est décrite par la formule suivante:

$$T_{\text{test}} = \text{MRP_TSTdefaultT} \times \text{MRP_TSTNRmax}$$

où

MRP_TSTdefaultT et MRP_TSTNRmax sont les paramètres conformes au Tableau 59.

La durée totale dont a besoin un MRM pour détecter une défaillance est non seulement T_{test} , mais également le temps mort entre l'instant précis où la défaillance s'est produite dans le réseau et la dernière réception d'une trame d'essai au niveau du MRM. L'instant précis de la défaillance peut se trouver n'importe quand entre le transfert de la dernière trame d'essai du MRM au point de défaillance avant la défaillance et l'instant prévu de l'arrivée de la trame d'essai suivante au point de défaillance, qui ne sera plus transmise. Comme on ne peut pas déterminer préalablement le point exact de la défaillance dans la topologie, la durée de transmission d'une trame d'essai MRP dans l'ensemble de l'anneau, T_{ring} doit être la base de calcul de ce temps mort:

$$T_{\text{ring}} = N \times (T_{\text{switch}} + T_{\text{queue}} + T_{\text{bit}} + T_{\text{line}})$$

où

N est le nombre total d'appareils en anneau participants;

T_{switch} est le délai introduit par chaque nœud de commutation;

T_{queue} est le délai introduit par une trame en cours de transmission vers le réseau, retardant ainsi le transfert de la trame d'essai du MRP, même si cette dernière est gérée dans une file d'attente de commutation à priorité élevée;

T_{bit} est la durée dont a besoin un appareil pour envoyer et recevoir le trafic d'essai du MRP (en général 5,12 μs à 100 Mbit/s Ethernet pour une trame de 64 octets);

T_{line} est le délai de propagation des trames sur le support physique. Si la distance entre les nœuds de commutation n'est pas exceptionnellement importante, cela peut être négligé.

La durée réelle entre l'occurrence de la défaillance et sa détection dans le MRM, T_{detect} , est donc calculée comme suit:

$$T_{\text{detect}} = T_{\text{test}} + T_{\text{ring}} \text{ ou}$$

$$T_{\text{detect}} = \text{MRP_TSTdefaultT} \times \text{MRP_TSTNRmax} + N \times (T_{\text{switch}} + T_{\text{queue}} + T_{\text{bit}} + T_{\text{line}}).$$

À la suite de la détection de la défaillance, le MRM commence à transmettre les trames de changement de topologie afin d'initier le vidage de la FDB dans les appareils en anneau après un temps d'immobilisation T_{hold} . Ce temps doit être ajouté au temps de reconfiguration total. Dans un scénario défavorable, le temps dont a besoin une trame de changement de topologie pour atteindre tous les appareils de l'anneau est également de T_{ring} , car la trame $\text{MRP_TopologyChange}$ analogue à la trame MRP_Test peut devoir traverser l'ensemble de l'anneau. Le temps dont a besoin le dernier appareil pour vider sa FDB T_{FDB} doit être ajouté, car l'anneau est uniquement opérationnel après que le dernier appareil a vidé sa base de données de filtrage, alors que tous les autres appareils l'ont fait plus tôt. Par conséquent, le temps nécessaire à l'anneau pour procéder à la reconfiguration est étendu du T_{flush} suivant:

$$T_{\text{flush}} = T_{\text{hold}} + T_{\text{ring}} + T_{\text{FDB}}$$

où

$T_{\text{hold}} = \text{MRP_TOPchgT} \times \text{MRP_TOPNRmax}$, issu du Tableau 61;

T_{FDB} est le temps dont a besoin un appareil pour vider sa FDB.

Pour le temps de reconfiguration total de l'anneau, T_{rec} , T_{detect} et T_{flush} doivent être ajoutés:

$T_{\text{rec}} = T_{\text{detect}} + T_{\text{flush}}$ ou

$T_{\text{rec}} = \text{MRP_TSTdefaultT} \times \text{MRPTSTNRmax} + 2 \times N \times (T_{\text{switch}} + T_{\text{queue}} + T_{\text{bit}} + T_{\text{line}}) + T_{\text{FDB}} + T_{\text{hold}}$

EXEMPLE Calcul avec l'ensemble cohérent de paramètres pour un temps de reprise de 10 ms:

$\text{MRP_TSTdefaultT} = 1 \text{ ms}$

$\text{MRP_TSTNRmax} = 3$

$N = 50$

$T_{\text{switch}} = 10 \text{ } \mu\text{s}$

$T_{\text{queue}} = 0$, en prenant pour hypothèse que les trames du MRP se trouvant dans la file d'attente à priorité élevée ne soient pas retardées par des trames se trouvant déjà dans la mémoire tampon d'un appareil en anneau et en cours de transmission sur le réseau.

$T_{\text{bit}} = 5,12 \text{ } \mu\text{s}$

$T_{\text{line}} = 0,5 \text{ } \mu\text{s}$ pour un câble de 100 m entre deux appareils réseau

$T_{\text{FDB}} = 500 \text{ } \mu\text{s}$

$T_{\text{hold}} = 0,5 \text{ ms} \times 3$, conformément à l'ensemble cohérent de paramètres pour 10 ms

Avec:

$T_{\text{rec}} = 1 \text{ ms} \times 3 + 2 \times 50 \times (10 \text{ } \mu\text{s} + 5,12 \text{ } \mu\text{s} + 0,5 \text{ } \mu\text{s}) + 0,5 \text{ ms} + 1,5 \text{ ms} \approx 6,6 \text{ ms}$

Il s'agit d'un calcul réalisé dans les cas les plus favorables. Les trames d'essai du MRP ne font, par hypothèse, l'objet d'aucun retard lors de la transmission des autres trames, le trafic du MRP étant, par hypothèse, transmis dans les meilleures circonstances possibles.

9.9.3 Calcul du scénario le plus défavorable pour un temps de reprise de 10 ms

Dans un scénario le moins favorable, chaque trame essentielle pour la reconfiguration de l'anneau fait l'objet d'un délai dans chaque appareil du réseau, introduit par une importante trame Ethernet de 1 522 octets, transférée avant la trame MRP. L'exemple de calcul doit donc présenter un facteur $T_{\text{queue}} = 122 \text{ } \mu\text{s}$ pour chaque appareil en anneau. Cela introduit deux contraintes de temporisation qui doivent être observées simultanément:

- La durée $T_{\text{ring}} + \text{MRP_TSTdefaultT}$ ne doit pas dépasser T_{test} . Si cela était le cas, un anneau ouvert pourrait être détecté par erreur dans le MRM, si le délai entre deux trames d'essai immédiatement envoyées l'une après l'autre varie fortement, en raison

d'un encombrement du réseau, par exemple. Cela signifie que T_{ring} pour le temps de reprise de 10 ms peut ne pas dépasser 2 ms.

- Le temps de reconfiguration total qui peut être obtenu dans l'anneau ne doit pas dépasser 10 ms.

EXEMPLE Calcul pour le scénario le plus défavorable avec l'ensemble cohérent de paramètres pour un temps de reprise de 10 ms:

$$MRP_TSTNR_{max} = 3$$

$$N = 14$$

$$T_{switch} = 10 \mu s$$

$T_{queue} = 122 \mu s$, en prenant pour hypothèse que les trames du MRP se trouvant dans la file d'attente à priorité élevée soient retardées par une trame de 1 522 octets déjà en mémoire tampon et en cours de transmission sur le réseau.

$$T_{queue} = 5,12 \mu s$$

$T_{line} = 0$, en prenant pour hypothèse que le retard de ligne est négligeable.

$$T_{FDB} = 500 \mu s$$

$$T_{hold} = 0,5 \text{ ms} \times 3 = 1,5 \text{ ms}$$

Avec:

$$T_{ring} = 14 \times (10 \mu s + 122 \mu s + 5,12 \mu s) \approx 1,9 \text{ ms}$$

$$T_{rec} = 1 \text{ ms} \times 3 + 2 \times 14 \times (10 \mu s + 122 \mu s + 5,12 \mu s) + 0,5 \text{ ms} + 1,5 \text{ ms} \approx 8,8 \text{ ms}$$

Cela montre que dans l'hypothèse des paramètres de temporisation ci-dessus et du scénario le plus défavorable, le temps de reprise de 10 ms peut être obtenu lorsque le nombre d'appareils dans l'anneau (N) est inférieur ou égal à 14.

9.9.4 Calcul du scénario le plus défavorable pour 50 appareils

Avec $N = 50$ appareils, l'ensemble cohérent de paramètres pour un temps de reprise de 30 ms et le même scénario le moins favorable que celui décrit en 9.9.3, le T_{ring} suivant peut être obtenu:

$$T_{ring} = 50 \times (10 \mu s + 122 \mu s + 5,12 \mu s) \approx 6,9 \text{ ms}$$

Si la trame $MRP_TSTdefaultT$ est prolongée à 3,5 ms, $T_{ring} + MRP_TSTdefaultT \approx 10,4 \text{ ms}$ est inférieur au seuil de détection d'un anneau ouvert dans le MRM, T_{test} , qui correspond à l'ensemble cohérent de paramètres pour 30 ms:

$$T_{test} = MRP_TSTdefaultT \times MRP_TSTNR_{max} = 3,5 \text{ ms} \times 3 = 10,5 \text{ ms}.$$

Cela donne lieu au temps de reconfiguration total de l'anneau ci-dessous pour 50 appareils dans le scénario le plus défavorable, avec l'ensemble cohérent de paramètres pour 30 ms:

$$T_{rec} = 3,5 \text{ ms} \times 3 + 2 \times 50 (10 \mu s + 122 \mu s + 5,12 \mu s) + 0,5 \text{ ms} + 1,5 \text{ ms} \approx 26,2 \text{ ms}$$

NOTE Pour assurer un moyen viable de calcul du temps de reprise réel d'un réseau, le fabricant de l'appareil revendiquant la prise en charge d'un temps de reprise de 10 ms et 30 ms peut indiquer T_{switch} et T_{FDB} des appareils de prise en charge.

9.10 Calcul du temps de sélection du gestionnaire automatique du MRP

Comme souligné en 5.10.3, le processus de sélection du gestionnaire est itératif. A chaque itération, deux gestionnaires automatiques sur trois prennent le rôle de client. Le nombre d'itérations n_{it} exigé pour sélectionner le meilleur gestionnaire dépend de l'ordre de priorité et du nombre d'appareils.

Chaque itération exige un maximum d'un cycle MRP_Test. Le temps nécessaire pour sélectionner le MRM dépend donc de l'ensemble de paramètres cohérent qu'il utilise, comme défini dans le Tableau 59.

Dans le cas d'un anneau composé de 50 gestionnaires automatiques, par exemple, qui utilise l'ensemble de paramètres cohérent de 200 ms du MRM, avec une répartition des priorités selon le scénario le plus défavorable, l'itération doit disposer de $n_{\text{it}} = 4$ étapes d'itération et la sélection du gestionnaire doit ainsi disposer du calcul suivant:

$$T_{\text{vot}} \approx \text{MRP_TSTdefaultT} \times n_{\text{it}} = 20 \text{ ms} \times 4 \approx 80 \text{ ms}$$

pour sélectionner parmi les MRM celui qui va finalement rester.

10 Base d'informations de gestion du MRP (MIB)

10.1 Généralités

Deux bases d'informations de gestion (MIB) de réseau facultatives permettent de configurer des nœuds du MRP

Le Paragraphe 10.2 présente une MIB avec une vue de surveillance. Elle peut être utilisée pour les appareils de terrain.

Le Paragraphe 10.3 présente une MIB avec une vue de gestion et de surveillance. Elle peut être utilisée pour les composants de réseau.

10.2 MIB du MRP avec vue de surveillance

NOTE Lors de la copie de la MIB à partir d'un document PDF tous les en-têtes et pieds de page sont supprimés et les caractères sont convertis au format exigé par le protocole SNMP. Bien que des précautions aient été prises afin de n'utiliser aucun caractère non valide, certains peuvent apparaître dans le document du fait du processus d'édition.

```
-- *****
-- Monitoring MIB definitions
-- *****

IEC- 62439-2-MIB DEFINITIONS ::= BEGIN

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

IMPORTS
    MODULE-IDENTITY,
    OBJECT-IDENTITY,
    OBJECT-TYPE,
    TimeTicks,
    Counter32,
    Unsigned32,
    Integer32          FROM SNMPv2-SMI
    OBJECT-GROUP,
    MODULE-COMPLIANCE FROM SNMPv2-CONF
```

```

TEXTUAL- CONVENTI ON,
DisplayString FROM SNMPv2- TC;
-- *****
-- Root OID
-- *****
iec62439 MODULE- I DENT I T Y
LAST- UPDATED "201405090000Z" -- May 09, 2014
ORGANI ZATI ON "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
"
DESCRIPTI ON "
    This MIB module defines the Network Management interfaces
    for the Redundancy Protocols defined by the IEC
    standard 62439.
    This definitions specify the pure monitoring variant
    of a SNMP entity. If the SNMP entity also contains
    management functionality, then IEC 62439 management MIB
    definitions shall be used instead of this MIB module.
"

REVISION "201405090000Z" -- May 09, 2014
DESCRIPTI ON "
    Manager (auto) support has been added in mrpDomainAdminRole.
    Interconnection functionality added.
    Units of mrpDomainRoundTripDelayMax and mrpDomainRoundTripDelayMin
    have been corrected to milliseconds.
    mrpDomainOperRole and mrpDomainAdminRole data type have been set
    to INTEGER.
    Missing constructs in import area have been added.
    Module identity block has been put on top.
"

REVISION "200811100000Z" -- November 10, 2008
DESCRIPTI ON "
    The undefined use of the value 'badValue' for objects
    of the optional 'mrpDomainDiagGroup' has been removed
    from DESCRIPTION clause at the objects concerned.
    The role dependency of the objects 'mrpDomainState' and
    'mrpDomainError' in the DESCRIPTION clause removed.
"

REVISION "200811100000Z" -- November 10, 2008
DESCRIPTI ON "
    Separation of IEC 62439 into a suite of documents
    This MIB applies to IEC 62439-2, no change in functionality
"

REVISION "200708240000Z" -- August 24, 2007
DESCRIPTI ON "
    Final Draft Proposal version of the Network Management
    interface for the Media Redundancy Protocol (MRP) with
    the division in pure Monitoring functionality and in
    additional management functionality.
"

::= { iso std(0) 62439 }

-- *****
-- Declaration of IEC62439UuidType
-- *****
IEC62439UuidType ::= TEXTUAL- CONVENTI ON
STATUS current
DESCRIPTI ON "
    The IEC 61158-5-10 defines the structure of
    the UUID as a data type numeric
    identifier 1025.
"
SYNTAX OCTET STRING (SIZE (16))

```

```

-- *****
-- Redundancy Protocols
-- *****
mrp          OBJECT IDENTIFIER ::= { iec62439 1 }
prp          OBJECT IDENTIFIER ::= { iec62439 2 }
crp          OBJECT IDENTIFIER ::= { iec62439 3 }
brp          OBJECT IDENTIFIER ::= { iec62439 4 }

-- *****
-- objects groups of redundancy object identifiers
-- *****
mrpObjects   OBJECT IDENTIFIER ::= { mrp 1 }
mrpNotifications OBJECT IDENTIFIER ::= { mrp 2 }
mrpConformance OBJECT IDENTIFIER ::= { mrp 3 }

-- *****
-- Objects of the MRP Network Management
-- *****
mrpDomainTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MrpDomainEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "
        Every entry in this table contains information about
        a Media Redundancy Protocol (MRP) domain.

        Entries cannot be created or deleted via SNMP
        operations.
    "
    ::= { mrpObjects 1 }

mrpDomainEntry OBJECT-TYPE
    SYNTAX      MrpDomainEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "An entry in the mrpDomainTable."
    INDEX       { mrpDomainIndex }
    ::= { mrpDomainTable 1 }

MrpDomainEntry ::= SEQUENCE {
    mrpDomainIndex      Unsigned32,
    mrpDomainID         IEC62439UuidType,
    mrpDomainName       DisplayString,
    mrpDomainAdminRole  INTEGER,
    mrpDomainOperRole   INTEGER,
    mrpDomainRingPort1  Integer32,
    mrpDomainRingPort1State INTEGER,
    mrpDomainRingPort2  Integer32,
    mrpDomainRingPort2State INTEGER,
    mrpDomainState       BITS,
    mrpDomainError       BITS,
    mrpDomainBlocked     INTEGER,
    mrpDomainVlanID      Unsigned32,
    mrpDomainManagerPriority INTEGER,
    mrpDomainRingOpenCount Counter32,
    mrpDomainLastRingOpenChange TimeTicks,
    mrpDomainRoundTripDelayMax Unsigned32,
    mrpDomainRoundTripDelayMin Unsigned32,
    mrpDomainResetRoundTripDelays INTEGER,
    mrpDomainMRMReactOnLinkChange INTEGER
}

mrpDomainIndex OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "The index of the entry"
    ::= { mrpDomainEntry 1 }

mrpDomainID OBJECT-TYPE
    SYNTAX      IEC62439UuidType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Universally unique identifier belongs to the MRP domain
        which represents a ring.
    "
    ::= { mrpDomainEntry 2 }

```

mrpDomainName	OBJECT- TYPE
SYNTAX	DisplayString
MAX- ACCESS	read-only
STATUS	current
DESCRIPTION	" A logical name for the MRP domain to ease the management of MRP domains. "
::= { mrpDomainEntry 3 }	
mrpDomainAdminRole	OBJECT- TYPE
SYNTAX	INTEGER { disable(0), client(1), manager(2), managerAutoComp(3), managerAuto(4) }
MAX- ACCESS	read-write
STATUS	current
DESCRIPTION	" Control the MRP behavior of the system per domain. If the value is set to disable(0) the MRP entity of this domain shall be disabled. If the value is set to client(1) the entity shall be set to the role of a Media Redundancy Client (MRC). If the value is set to manager(2) the entity shall be set to the role of a Media Redundancy Manager (MRM). If the value is set to managerAutoComp(3) the entity shall be set to the role of a Media Redundancy Manager Auto (MRA) complying to Annex A. If the value is set to managerAuto(4) the entity shall be set to the role of a Media Redundancy Manager Auto (MRA) not supporting Annex A. The factory settings are recommended to adjust the value of this object to the client(1) capability of the component, or, if supported, to the managerAutoComp(3) or managerAuto(4) capability. in order to prevent multiple managers are in ring (the order of the capabilities are not necessarily conform to the order of the object values here). If the agent restricts the write access, no matter what reason, it shall reject write requests by responding with 'badValue'. "
::= { mrpDomainEntry 4 }	
mrpDomainOperRole	OBJECT- TYPE
SYNTAX	INTEGER { disable(0), client(1), manager(2), managerAutoComp(3), managerAuto(4) }
MAX- ACCESS	read-only
STATUS	current
DESCRIPTION	" The operational role of an MRP entity per domain. A value of disabled(0) signals that the entity doesn't work (whatever reason). A value of client(1) signals that the entity is in a client role. A value of manager(2) signals that the entity is the manager of this MRP domain. A value of managerAutoComp(3) signals that the entity is in automanager role complying to Annex A. A value of managerAuto(4) signals that the entity is in automanager role not supporting Annex A. "
::= { mrpDomainEntry 5 }	
mrpDomainRingPort1	OBJECT- TYPE
SYNTAX	Integer32
MAX- ACCESS	read-only

```

STATUS          current
DESCRIPTION      "
                  The ifIndex of the layer 2 interface which is used
                  as ring port 1.
                  "
::= { mrpDomainEntry 6 }

mrpDomainRingPort1State OBJECT-TYPE
SYNTAX          INTEGER {
                  disabled(1),
                  blocked(2),
                  forwarding(3),
                  not-connected(4)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION      "
                  Operational state of the first Ring-Port.

                  disabled(1) indicates that all frames are dropped.
                  blocked(2) indicates that all frames are dropped except
                              – MRP topology change frames and MRP test frames from
                                a MRM,
                              – MRP link change frames from an MRC,
                              – MRP interconnection topology change from a MIM,
                              – MRP interconnection link change from a MIC,
                              – frames from other protocols that also define to pass
                                blocked(2) ports.
                  forwarding(3) indicates that all frames are passed through
                              according to the forwarding behavior
                              of IEEE 802.1D.
                  not-connected(4) indicates that the port has no link.
                  "
::= { mrpDomainEntry 7 }

mrpDomainRingPort2 OBJECT-TYPE
SYNTAX          Integer32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION      "
                  The ifIndex of the layer 2 interface which is
                  used as ring port 2.
                  "
::= { mrpDomainEntry 8 }

mrpDomainRingPort2State OBJECT-TYPE
SYNTAX          INTEGER {
                  disabled(1),
                  blocked(2),
                  forwarding(3),
                  not-connected(4)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION      "
                  Operational state of the second Ring-Port.

                  disabled(1) indicates that all frames are dropped.
                  blocked(2) indicates that all frames are dropped except
                              – MRP topology change frames and MRP test frames from
                                a MRM,
                              – MRP link change frames from an MRC,
                              – MRP interconnection topology change from a MIM,
                              – MRP interconnection link change from a MIC,
                              – frames from other protocols that also define to pass
                                blocked(2) ports.
                  forwarding(3) indicates that all frames are passed through
                              according to the forwarding behavior
                              of IEEE 802.1D.
                  not-connected(4) indicates that the port has no link.
                  "
::= { mrpDomainEntry 9 }

mrpDomainState OBJECT-TYPE
SYNTAX          BITS {
                  disabled(0),
                  undefined(1),

```

```

        ringOpen(2),
        reserved(3)
    }
MAX- ACCESS      read-only
STATUS           current
DESCRIPTION      "
    Operational status of the MRP entity.
    disabled(0) – MRP switched off. All higher bits are invalid
                  and shall be reset.
    undefined(1) – Value is not valid.
                  All higher bits are invalid and shall be reset.
    ringOpen(2) – MRP ring redundancy lost
                  All higher bits are invalid and shall be reset.
    reserved(3) – reserved for further extensions.
"
 ::= {   mrpDomainEntry 10   }

mrpDomainError OBJECT-TYPE
SYNTAX          BITS {
    noError(0),
    invalidVlanId(1),
    invalid(2),
    multipleMRM(3),
    singleSideReceive(4)
}
MAX- ACCESS      read-only
STATUS           current
DESCRIPTION      "
    If the device couldn't switch to the desired administrative state
    (thus the value of 'mrpDomainState' is not the expected one),
    this value provides the reason. Otherwise the bit noError(0) is set.

    noError(0)          – the operational state of the device is conform
                        to administrative state.
                        All higher bits are invalid and shall be reset.
    invalidVlanId(1)    – the assigned VLAN ID is not permitted .
    invalid(2)          – Value is not valid.
                        All higher bits are invalid and shall be reset.
    multipleMRM(3)      – multiple active managers in ring domain.
    singleSideReceive(4) – the test frames of an MRM
                        have been seen, but only on one port.
"
 ::= {   mrpDomainEntry 11   }

mrpDomainBlocked OBJECT-TYPE
SYNTAX          INTEGER {
    enabled(1),
    disabled(2)
}
MAX- ACCESS      read-only
STATUS           current
DESCRIPTION      "
    The entity supports blocked ring ports.
    Shows whether a mrp domain requires the support of the
    BLOCKED port state at ring ports.
    The manager of a mrp domain decides whether this support
    is required.
    Set to enabled the manager demands that all clients
    shall support the blocked attribute also set to enabled.
    If mrpDomainBlocked is set disabled at the manager,
    then the value of mrpDomainBlocked can be arbitrary
    at the clients.

    enabled(1) client: supports ring ports whose port state can be
                    blocked.
                manager: works only with clients supporting blocked
                    ring ports.
    disabled(2) client: no support of blocked ring ports.
                manager: Work with clients supporting blocked ring
                    ports and with clients not supporting
                    blocked ring ports.
"
 ::= {   mrpDomainEntry 12   }

mrpDomainVlanId OBJECT-TYPE
SYNTAX          Unsigned32
MAX- ACCESS      read-only

```

```

STATUS          current
DESCRIPTION     "
                The VLAN ID assigned to the MRP protocol.
                The VLAN ID only is in use when the bit invalidVlanId
                is not set in mrpDomainError.
                If value is set to 0 no VLAN is assigned.
                The invalidVlanId shall be set to 0 (no Error).
                "
 ::= { mrpDomainEntry 13 }

-- *****
-- Objects of the MRP Network Management for manager role
-- *****
mrpDomainManagerPriority OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION  "
                The priority of this MRP entity. If the device is
                client only, the value of this object shall be ignored
                by the MRP entity. Only the four most significant bits
                shall be used, the bits 0 to 11 are reserved. The
                smaller value has the higher priority.
                "
 ::= { mrpDomainEntry 14 }

mrpDomainRingOpenCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION  "
                Counter for ring-state changes to open.
                "
 ::= { mrpDomainEntry 15 }

mrpDomainLastRingOpenChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION  "
                Timeticks since last change of ring-state
                to ring open.
                "
 ::= { mrpDomainEntry 16 }

mrpDomainRoundTripDelayMax OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION  "
                The max. Round-Trip-Delay (in milliseconds)
                which was measured since startup.
                "
 ::= { mrpDomainEntry 17 }

mrpDomainRoundTripDelayMin OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION  "
                The min. Round-Trip-Delay (in milliseconds)
                which was measured since startup.
                "
 ::= { mrpDomainEntry 18 }

mrpDomainResetRoundTripDelays OBJECT-TYPE
    SYNTAX      INTEGER {
                    resetDelays(1)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION  "
                A write request with resetDelays(1) shall reset
                the values of mrpDomainRoundTripDelayMax and
                mrpDomainRoundTripDelayMin to zero at the
                same time.
    
```

```

"
 ::= { mrpDomainEntry 19 }

mrpDomainMRMReactOnLinkChange OBJECT-TYPE
    SYNTAX      INTEGER {
        enabled(1),
        disabled(2)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
        Tells whether the MRM reacts on link down MRP link change
        frames.
        enabled(1) indicates that the MRM reacts immediately on link down MRP
        link change frames.
        disabled(2) indicates that the MRM does not react on
        link down MRP link change frames.
    "
 ::= { mrpDomainEntry 20 }

-- *****
-- Objects of the Media redundancy Interconnection
-- *****
mrpInterconnectionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MrpInterconnectionEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION   "
        Every entry in this table contains information about
        a Media redundancy Interconnection domain.

        Entries cannot be created or deleted via SNMP
        operations.
    "
 ::= { mrpObjects 2 }

mrpInterconnectionEntry OBJECT-TYPE
    SYNTAX      MrpInterconnectionEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION   "An entry in the mrpInterconnectionTable."
    INDEX       { mrpInterconnectionID }
 ::= { mrpInterconnectionTable 1 }

MrpInterconnectionEntry ::= SEQUENCE {
    mrpInterconnectionID      Unsigned32,
    mrpInterconnectionName    DisplayString,
    mrpInterconnectionPort    Integer32,
    mrpInterconnectionPortState INTEGER,
    mrpInterconnectionVlanID  Unsigned32,
    mrpInterconnectionMode    INTEGER,
    mrpInterconnectionAdminRole INTEGER,
    mrpInterconnectionOperRole INTEGER,
    mrpInterconnectionState   INTEGER,
    mrpInterconnectionError   BITS,
    mrpInterconnectionOpenCount Counter32,
    mrpInterconnectionLastOpenChange TimeTicks,
    mrpInterconnectionPeerMAC  OCTET STRING,
    mrpInterconnectionOwnMRMMAC OCTET STRING,
    mrpInterconnectionPeerMRMMAC OCTET STRING,
    mrpInterconnectionPeerRingDomainUUID IEC62439UuidType
}

mrpInterconnectionID OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION   "The index of the entry"
 ::= { mrpInterconnectionEntry 1 }

mrpInterconnectionName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
        A logical name for the Media redundancy Interconnection domain to
        ease the management of Media redundancy Interconnection domains.
    "

```

```

 ::= { mrpInterconnectionEntry 2 }

mrpInterconnectionPort OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "
        The ifIndex of the layer 2 interface which is used
        as interconnection port.
        "
 ::= { mrpInterconnectionEntry 3 }

mrpInterconnectionPortState OBJECT-TYPE
    SYNTAX      INTEGER {
        disabled(1),
        blocked(2),
        forwarding(3),
        notConnected(4)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "
        Operational state of the Interconnection port.

        disabled(1) indicates that all frames are dropped.
        blocked(2) indicates that all frames are dropped except
        – MRP Interconnection topology change frames,
        – MRP Interconnection test frames,
        – MRP Interconnection link change frames,
        – MRP Interconnection link status poll frames,
        – frames from other protocols that also define to pass
          blocked(2) ports.
        forwarding(3) indicates that all frames are passed through
          according to the forwarding behavior of IEEE 802.1D.
        notConnected(4) indicates that the port has no link.
        "
 ::= { mrpInterconnectionEntry 4 }

mrpInterconnectionVlanId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "
        The VLAN ID assigned to the MRP Interconnection protocol.
        The VLAN ID only is in use when mrpInterconnectionError
        does not signal invalidVlanId.
        If value is set to 0 no VLAN is assigned.
        "
 ::= { mrpInterconnectionEntry 5 }

mrpInterconnectionMode OBJECT-TYPE
    SYNTAX      INTEGER {
        undefined(1),
        lcMode(2),
        rcMode(3)
    }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "
        Control the MRP Interconnection mode.

        The value undefined(1) signals that the interconnection mode is
        neither set to lcMode(2) nor to rcMode(3).
        If the value is set to lcMode(2) the device in Media redundancy
        Interconnection Manager (MI M) role is using the link check mode to
        detect open or closed interconnection links.
        If the value is set to rcMode(3) the device in Media redundancy
        Interconnection Manager (MI M) role is using the ring check mode to
        detect open or closed interconnection links.

        The factory settings are recommended to adjust the
        value of this object to the lcMode(2) capability.
        If the agent restricts the write access, no matter what
        reason, it shall reject write requests by responding
        with 'badValue'.
        "
 ::= { mrpInterconnectionEntry 6 }

```

```

mrpInterconnectionAdminRole OBJECT-TYPE
    SYNTAX      INTEGER {
        disable(0),
        client(1),
        manager(2)
    }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION   "
        Control the MRP Interconnection behavior of the system per
        Media redundancy Interconnection domain.

        If the value is set to disable(0) the MRP Interconnection
        entity of this domain shall be disabled.
        If the value is set to client(1) the entity shall be set
        to the role of a Media redundancy Interconnection Client (MIC).
        If the value is set to manager(2) the entity shall be set
        to the role of a Media redundancy Interconnection Manager (MIM).
        The factory settings are recommended to adjust the
        value of this object to the client(1) capability of the
        component in order to prevent multiple managers are
        in the interconnection (the order of the capabilities are not
        necessarily conform to the order of the object values
        here).
        If the agent restricts the write access, no matter what
        reason, it shall reject write requests by responding
        with 'badValue'.
    "

    ::= { mrpInterconnectionEntry 7 }

mrpInterconnectionOperRole OBJECT-TYPE
    SYNTAX      INTEGER {
        disable(0),
        client(1),
        manager(2)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
        The operational role of an MRP Interconnection entity
        per Media redundancy Interconnection domain.
        A value of disabled(0) signals that the entity doesn't
        work (whatever reason).
        A value of client(1) signals that the entity is in a
        Media redundancy Interconnection Client role.
        A value of manager(2) signals that the entity is the
        manager of this Media redundancy Interconnection domain.
    "

    ::= { mrpInterconnectionEntry 8 }

mrpInterconnectionState OBJECT-TYPE
    SYNTAX      INTEGER {
        disable(0),
        undefined(1),
        interconnOpen(2),
        interconnClosed(3)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
        Operational status of the MRP Interconnection entity.
        disable(0)      - The entity is disabled.
        undefined(1)    - Value is not valid.
        interconnOpen(2) - MRP Interconnection redundancy lost of this Media
                           redundancy Interconnection domain.
        interconnClosed(3) - MRP Interconnection redundancy available of this Media
                           redundancy Interconnection domain.
    "

    ::= { mrpInterconnectionEntry 9 }

mrpInterconnectionError OBJECT-TYPE
    SYNTAX      BITS {
        noError(0),
        invalidVlanId(1)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "

```

Configuration status of the MRP Interconnection entity.

```

noError(0)           –no configuration error detected.
invalidVlanId(1) –the assigned VLAN ID is not permitted.
"
 ::= {   mrpInterconnectionEntry 10 }

mrpInterconnectionOpenCount  OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
                Counter for MRP Interconnection state changes to
                interconnOpen.
                "
 ::= {   mrpInterconnectionEntry 11 }

mrpInterconnectionLastOpenChange  OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
                Timeticks since last change of
                mrpInterconnectionState to interconnOpen.
                "
 ::= {   mrpInterconnectionEntry 12 }

mrpInterconnectionPeerMAC  OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
                Device-MAC-Address of the devices interconnection peer, connected
                to the devices interlink port. This variable contains
                the MAC-Address contained in the media link check frame received at the
                interconnection port.
                "
 ::= {   mrpInterconnectionEntry 13 }

mrpInterconnectionOwnMRMMAC  OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
                The device-MAC-Address of the devices ring manager (MRM).
                This variable contains the MAC-Address extracted from the
                periodically received MRP_Test frame sent by the ring manager.
                If the ring managers MAC-Address is not known, the variable shall
                contain ff-ff-ff-ff-ff-ff.
                "
 ::= {   mrpInterconnectionEntry 14 }

mrpInterconnectionPeerMRMMAC  OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
                Device-MAC-Address of devices interconnection peer's, connected
                to the devices interlink port, ring manager.
                This variable contains the MRM- MAC-Address of the peers ring manager
                contained in the media link check frame received at the interconnection port.
                If the ring managers MAC-Address is not known, the variable shall
                contain ff-ff-ff-ff-ff-ff.
                "
 ::= {   mrpInterconnectionEntry 15 }

mrpInterconnectionPeerRingDomainUUID  OBJECT-TYPE
    SYNTAX      IEC62439UuidType
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "
                The domain UUID of the ring the interconnection peer's, connected
                to the devices interlink port, ring.
                This variable contains the MRM_DomainUUID of the peers ring
                contained in the media link check frame received at the interconnection port.
                "
 ::= {   mrpInterconnectionEntry 16 }

-- *****
-- conformance statements
-- *****
mrpRedundancyGroups  OBJECT IDENTIFIER ::= { mrpConformance 1 }

```

```

mrpDomainBasicGroup OBJECT- GROUP
  OBJECTS {
    mrpDomainIndex,
    mrpDomainID,
    mrpDomainName,
    mrpDomainAdminRole,
    mrpDomainOperRole,
    mrpDomainRingPort1,
    mrpDomainRingPort1State,
    mrpDomainRingPort2,
    mrpDomainRingPort2State,
    mrpDomainState,
    mrpDomainBlocked
  }
  STATUS current
  DESCRIPTION "
    Parameters which are necessary to monitor status of
    MRP domains.
  "

  ::= { mrpRedundancyGroups 1 }

mrpDomainManagerGroup OBJECT- GROUP
  OBJECTS {
    mrpDomainManagerPriority
  }
  STATUS current
  DESCRIPTION "
    Parameters which are additional to monitor status of
    MRP domain managers.
  "

  ::= { mrpRedundancyGroups 2 }

mrpDomainBlockingCntlGroup OBJECT- GROUP
  OBJECTS {
    mrpDomainMRMReactOnLinkChange
  }
  STATUS current
  DESCRIPTION "
    Parameters which are optional to monitor and
    control blocking behavior of MRP domains.
  "

  ::= { mrpRedundancyGroups 3 }

mrpDomainDiagGroup OBJECT- GROUP
  OBJECTS {
    mrpDomainError,
    mrpDomainRingOpenCount,
    mrpDomainLastRingOpenChange,
    mrpDomainRoundTripDelayMax,
    mrpDomainRoundTripDelayMin,
    mrpDomainResetRoundTripDelays
  }
  STATUS current
  DESCRIPTION "
    Parameters which are optional to monitor
    and control the diagnostic status of MRP domains.
  "

  ::= { mrpRedundancyGroups 4 }

mrpDomainAdvancedGroup OBJECT- GROUP
  OBJECTS {
    mrpDomainVlanId
  }
  STATUS current
  DESCRIPTION "
    Parameters which are optional to monitor
    the status of MRP domains.
  "

  ::= { mrpRedundancyGroups 5 }

mrpInterconnectionBasicGroup OBJECT- GROUP
  OBJECTS {
    mrpInterconnectionName,
    mrpInterconnectionPort,
    mrpInterconnectionPortState,
    mrpInterconnectionAdminRole,
    mrpInterconnectionOperRole,
    mrpInterconnectionMode,
  }

```

```

        mrpInterconnectionState
    }
    STATUS          current
    DESCRIPTION     "
                    Parameters which are necessary to monitor
                    the status of Media redundancy
                    Interconnection domains.
                    "
    ::= { mrpRedundancyGroups 6 }

mrpInterconnectionDiagGroup OBJECT-GROUP
    OBJECTS {
        mrpInterconnectionError,
        mrpInterconnectionOpenCount,
        mrpInterconnectionLastOpenChange,
        mrpInterconnectionPeerMAC,
        mrpInterconnectionOwnMRMMAC,
        mrpInterconnectionPeerMRMMAC,
        mrpInterconnectionPeerRingDomainUID
    }
    STATUS          current
    DESCRIPTION     "
                    Parameters which are optional to monitor
                    the diagnostic status of Media
                    redundancy Interconnection domains.
                    "
    ::= { mrpRedundancyGroups 7 }

mrpInterconnectionAdvancedGroup OBJECT-GROUP
    OBJECTS {
        mrpInterconnectionVlanId
    }
    STATUS          current
    DESCRIPTION     "
                    Parameters which are optional to monitor the
                    status of Media redundancy Interconnection domains.
                    "
    ::= { mrpRedundancyGroups 8 }

-- *****
-- compliance specifications
-- *****
mrpRedundancyCompliances OBJECT IDENTIFIER ::= { mrpConformance 2 }

mrpDomainBasicCompliance MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION     "
                    Basic implementation requirements for MRP support.
                    The agent shall support the monitoring of redundancy
                    domains.
                    "

    MODULE
    MANDATORY-GROUPS {
        mrpDomainBasicGroup
    }

    GROUP mrpDomainAdvancedGroup
    DESCRIPTION     "
                    For SNMP agents it is optional to support
                    advanced monitoring of redundancy domains.
                    "

    ::= { mrpRedundancyCompliances 1 }

mrpDomainManagerCompliance MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION     "
                    Manager implementation requirements for MRP
                    support. The agent shall support the monitoring
                    of redundancy domains.
                    "

    MODULE
    MANDATORY-GROUPS {
        mrpDomainBasicGroup,
        mrpDomainManagerGroup
    }

    GROUP mrpDomainBlockingCtrlGroup
    DESCRIPTION     "
                    For SNMP agents it is optional to support blocking control.

```

```

"
GROUP      mrpDomainDiagGroup
DESCRIPTION "
    For SNMP agents it is optional to support
    monitoring of the diagnostic status of MRP domains.
"

::= { mrpRedundancyCompliances 2 }

mrpInterconnectionCompliance  MODULE- COMPLIANCE
STATUS      current
DESCRIPTION "
    Implementation requirements for MRP Interconnection support.
    The agent shall support the monitoring of
    Media redundancy Interconnection domains.
"

MODULE
MANDATORY- GROUPS {
    mrpInterconnectionBasicGroup
}

GROUP      mrpInterconnectionAdvancedGroup
DESCRIPTION "
    For SNMP agents it is optional to support
    advanced monitoring of Media redundancy
    Interconnection domains.
"

GROUP      mrpInterconnectionDiagGroup
DESCRIPTION "
    For SNMP agents it is optional to support
    monitoring of the diagnostic status of Media
    redundancy Interconnection domains.
"

::= { mrpRedundancyCompliances 3 }

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

```

10.3 MIB du MRP avec vue de gestion et de surveillance

NOTE Lors de la copie de la MIB à partir d'un document PDF tous les en-têtes et pieds de page sont supprimés et les caractères sont convertis au format exigé par le protocole SNMP. Bien que des précautions aient été prises afin de n'utiliser aucun caractère non valide, certains peuvent apparaître dans le document du fait du processus d'édition.

```

-- *****
--      Management MIB definitions
-- *****

IEC- 62439- 2- MIB          DEFINITIONS ::= BEGIN

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

IMPORTS
    MODULE- IDENTIFICATION,
    OBJECT- IDENTIFICATION,
    OBJECT- TYPE,
    TimeTicks,
    Counter32,
    Unsigned32,
    Integer32                FROM SNMPv2- SMI
    OBJECT- GROUP,
    MODULE- COMPLIANCE       FROM SNMPv2- CONF
    TEXTUAL- CONVENTION,
    DisplayString, RowStatus FROM SNMPv2- TC;
-- *****
--      Root OID
-- *****
iec62439  MODULE- IDENTIFICATION
    LAST- UPDATED "201405090000Z" - - May 09, 2014
    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.

This definitions specify the monitoring and management variant of a SNMP entity. If the SNMP entity only contains monitoring functionality, then IEC 62439 monitoring MIB definitions shall be used instead of this MIB module.

An entity may support either the monitoring facet only or the management facet including the monitoring facet as a subset.

"

REVISION "201405090000Z" -- May 09, 2014

DESCRIPTION "

Manager (auto) support has been added in mrpDomainAdminRole. Interconnection functionality added.

Units of mrpDomainRoundTripDelayMax and mrpDomainRoundTripDelayMin have been corrected to milliseconds.

mrpDomainOperRole and mrpDomainAdminRole data type have been set to INTEGER.

Missing constructs in import area have been added.

Module identity block has been put on top.

"

REVISION "200711120000Z" -- November 12, 2007

DESCRIPTION "

The undefined use of the value 'badValue' for objects of the optional 'mrpDomainDiagGroup' has been removed from DESCRIPTION clause at the objects concerned.

The role dependency of the objects 'mrpDomainState' and 'mrpDomainError' in the DESCRIPTION clause removed.

"

REVISION "200708240000Z" -- August 24, 2007

DESCRIPTION "

Final Draft Proposal version of the Network Management interface for the Media Redundancy Protocol (MRP) with the monitoring functionality and the additional management functionality.

"

::= { iso std(0) 62439 }

-- *****

-- Declaration of IEC62439UuidType

-- *****

IEC62439UuidType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION "

The IEC 61158-5-10 defines the structure of the UUID as a data type numeric identifier 1025.

"

SYNTAX OCTET STRING (SIZE (16))

-- *****

-- Redundancy Protocols

-- *****

mrp OBJECT IDENTIFIER ::= { iec62439 1 }

prp OBJECT IDENTIFIER ::= { iec62439 2 }

crp OBJECT IDENTIFIER ::= { iec62439 3 }

brp OBJECT IDENTIFIER ::= { iec62439 4 }

-- *****

-- objects groups of redundancy object identifiers

-- *****

mrpObjects OBJECT IDENTIFIER ::= { mrp 1 }

mrpNotifications OBJECT IDENTIFIER ::= { mrp 2 }

```

mrpConformance OBJECT IDENTIFIER ::= { mrp 3 }

-- *****
-- Objects of the MRP Network Management
-- *****
mrpDomainTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MRPDomainEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "
        Every entry in this table contains information about
        a Media Redundancy Protocol (MRP) domain.

        Row entries can be created or deleted via
        SNMP operations. Therefore the MAX-ACCESS values
        of some Object Identifier are different from the
        values of the same Object Identifier in the
        pure monitoring facet.
        "
    ::= { mrpObjects 1 }

mrpDomainEntry OBJECT-TYPE
    SYNTAX      MRPDomainEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the mrpDomainTable."
    INDEX      { mrpDomainIndex }
    ::= { mrpDomainTable 1 }

MRPDomainEntry ::= SEQUENCE {
    mrpDomainIndex      Unsigned32,
    mrpDomainID         IEC62439UuidType,
    mrpDomainName       DisplayString,
    mrpDomainAdminRole  INTEGER,
    mrpDomainOperRole   INTEGER,
    mrpDomainRingPort1  Integer32,
    mrpDomainRingPort1State  INTEGER,
    mrpDomainRingPort2  Integer32,
    mrpDomainRingPort2State  INTEGER,
    mrpDomainState      BITS,
    mrpDomainError      BITS,
    mrpDomainBlocked    INTEGER,
    mrpDomainVlanID     Unsigned32,
    mrpDomainManagerPriority  INTEGER,
    mrpDomainRingOpenCount  Counter32,
    mrpDomainLastRingOpenChange  TimeTicks,
    mrpDomainRoundTripDelayMax  Unsigned32,
    mrpDomainRoundTripDelayMin  Unsigned32,
    mrpDomainResetRoundTripDelays  INTEGER,
    mrpDomainMRMReactOnLinkChange  INTEGER,
    mrpDomainRowStatus      RowStatus
}

mrpDomainIndex OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index of the entry"
    ::= { mrpDomainEntry 1 }

mrpDomainID OBJECT-TYPE
    SYNTAX      IEC62439UuidType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "
        Universally unique identifier belongs to the MRP domain
        which represents a ring.
        "
    ::= { mrpDomainEntry 2 }

mrpDomainName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "
        A logical name for the MRP domain to ease the
        management of MRP domains.
        "

```

::= { mrpDomainEntry 3 }

mrpDomainAdminRole	OBJECT- TYPE
SYNTAX	INTEGER { disable(0), client(1), manager(2), managerAutoComp(3), managerAuto(4) }
MAX- ACCESS	read-write
STATUS	current
DESCRIPTION	" Control the MRP behavior of the system per domain. If the value is set to disable(0) the MRP entity of this domain shall be disabled. If the value is set to client(1) the entity shall be set to the role of a Media Redundancy Client (MRC). If the value is set to manager(2) the entity shall be set to the role of a Media Redundancy Manager (MRM). If the value is set to managerAutoComp(3) the entity shall be set to the role of a Media Redundancy Manager Auto (MRA) complying to Annex A. If the value is set to managerAuto(4) the entity shall be set to the role of a Media Redundancy Manager Auto (MRA) not supporting Annex A. The factory settings are recommended to adjust the value of this object to the client(1) capability of the component, or, if supported, to the managerAutoComp(3) or managerAuto(4) capability, in order to prevent multiple managers are in ring (the order of the capabilities are not necessarily conform to the order of the object values here). If the agent restricts the write access, no matter what reason, it shall reject write requests by responding with 'badValue'. "

::= { mrpDomainEntry 4 }

mrpDomainOperRole	OBJECT- TYPE
SYNTAX	INTEGER { disable(0), client(1), manager(2), managerAutoComp(3), managerAuto(4) }
MAX- ACCESS	read-only
STATUS	current
DESCRIPTION	" The operational role of an MRP entity per domain. A value of disabled(0) signals that the entity doesn't work (whatever reason). A value of client(1) signals that the entity is in a client role. A value of manager(2) signals that the entity is the manager of this MRP domain. A value of managerAutoComp(3) signals that the entity is in automanager role complying to Annex A. A value of managerAuto(4) signals that the entity is in automanager role not supporting Annex A. "

::= { mrpDomainEntry 5 }

mrpDomainRingPort1	OBJECT- TYPE
SYNTAX	Integer32
MAX- ACCESS	read-create
STATUS	current
DESCRIPTION	" The ifIndex of the layer 2 interface which is used as ring port 1. "

::= { mrpDomainEntry 6 }

mrpDomainRingPort1State	OBJECT- TYPE
SYNTAX	INTEGER {

```

disabled(1),
blocked(2),
forwarding(3),
not-connected(4)
}
MAX- ACCESS      read-only
STATUS           current
DESCRIPTION      "
                  Operational state of the first Ring-Port.

disabled(1)       indicates that all frames are dropped.
blocked(2)       indicates that all frames are dropped except
                  – MRP topology change frames and MRP test frames from
                    a MRM,
                  – MRP link change frames from an MRC,
                  – MRP interconnection topology change from a MIM,
                  – MRP interconnection link change from a MIC,
                  – frames from other protocols that also define to pass
                    blocked(2) ports.
forwarding(3)     indicates that all frames are passed through
                  according to the forwarding behavior
                  of IEEE 802.1D.
not-connected(4) indicates that the port has no link.
"
::= { mrpDomainEntry 7 }

mrpDomainRingPort2 OBJECT-TYPE
SYNTAX            Integer32
MAX- ACCESS      read-create
STATUS           current
DESCRIPTION      "
                  The ifIndex of the layer 2 interface which is
                  used as ring port 2.
                  "
::= { mrpDomainEntry 8 }

mrpDomainRingPort2State OBJECT-TYPE
SYNTAX            INTEGER {
                  disabled(1),
                  blocked(2),
                  forwarding(3),
                  not-connected(4)
                  }
MAX- ACCESS      read-only
STATUS           current
DESCRIPTION      "
                  Operational state of the second Ring-Port.

disabled(1)       indicates that all frames are dropped.
blocked(2)       indicates that all frames are dropped except
                  – MRP topology change frames and MRP test frames from
                    a MRM,
                  – MRP link change frames from an MRC,
                  – MRP interconnection topology change from a MIM,
                  – MRP interconnection link change from a MIC,
                  – frames from other protocols that also define to pass
                    blocked(2) ports.
forwarding(3)     indicates that all frames are passed through
                  according to the forwarding behavior
                  of IEEE 802.1D.
not-connected(4) indicates that the port has no link.
"
::= { mrpDomainEntry 9 }

mrpDomainState OBJECT-TYPE
SYNTAX            BITS {
                  disabled(0),
                  undefined(1),
                  ringOpen(2),
                  reserved(3)
                  }
MAX- ACCESS      read-only
STATUS           current
DESCRIPTION      "
                  Operational status of the MRP entity.
                  disabled(0) – MRP switched off. All higher bits are invalid
                              and shall be reset.

```

```

    undefined(1) – Value is not valid.
                      All higher bits are invalid and shall be reset.
    ringOpen(2) – MRP ring redundancy lost
                      All higher bits are invalid and shall be reset.
    reserved(3) – reserved for further extensions.

    "
 ::= {   mrpDomainEntry 10   }

mrpDomainError OBJECT- TYPE
SYNTAX         BITS {
                noError(0),
                invalidVlanId(1),
                invalid(2),
                multipleMRM(3),
                singleSideReceive(4)
                }
MAX- ACCESS    read-only
STATUS         current
DESCRIPTION    "
                If the device couldn't switch to the desired administrative state
                (thus the value of 'mrpDomainState' is not the expected one),
                this value provides the reason. Otherwise the bit noError(0) is set.

    noError(0)      –the operational state of the device is conform
                      to administrative state.
                      All higher bits are invalid and shall be reset.
    invalidVlanId(1) –the assigned VLAN ID is not permitted .
    invalid(2)      –Value is not valid.
                      All higher bits are invalid and shall be reset.
    multipleMRM(3)  –multiple active managers in ring domain.
    singleSideReceive(4) –the test frames of an MRM
                      have been seen, but only on one port.

    "
 ::= {   mrpDomainEntry 11   }

mrpDomainBlocked OBJECT- TYPE
SYNTAX           INTEGER {
                enabled(1),
                disabled(2)
                }
MAX- ACCESS      read-only
STATUS           current
DESCRIPTION      "
                The entity supports blocked ring ports.
                Shows whether a mrp domain requires the support of the
                BLOCKED port state at ring ports.
                The manager of a mrp domain decides whether this support
                is required.
                Set to enabled the manager demands that all clients
                shall support the blocked attribute also set to enabled.
                If mrpDomainBlocked is set disabled at the manager,
                then the value of mrpDomainBlocked can be arbitrary
                at the clients.

    enabled(1)     client: supports ring ports whose port state can be
                      blocked.
                      manager: works only with clients supporting blocked
                      ring ports.
    disabled(2)    client: no support of blocked ring ports.
                      manager: Work with clients supporting blocked ring
                      ports and with clients not supporting
                      blocked ring ports.

    "
 ::= {   mrpDomainEntry 12   }

mrpDomainVlanId OBJECT- TYPE
SYNTAX           Unsigned32
MAX- ACCESS      read-create
STATUS           current
DESCRIPTION      "
                The VLAN ID assigned to the MRP protocol.
                The VLAN ID only is in use when the bit invalidVlanId
                is not set in mrpDomainError.
                If value is set to 0 no VLAN is assigned.
                The invalidVlanId shall be set to 0 (no Error).

    "
 ::= {   mrpDomainEntry 13   }

```

```
-- *****
-- Objects of the MRP Network Management for manager role
-- *****

mrpDomainManagerPriority OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
        The priority of this MRP entity. If the device is
        client only, the value of this object shall be ignored
        by the MRP entity. Only the four most significant bits
        shall be used, the bits 0 to 11 are reserved. The
        smaller value has the higher priority.
    "

    ::= { mrpDomainEntry 14 }

mrpDomainRingOpenCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Counter for ring-state changes to open.
    "

    ::= { mrpDomainEntry 15 }

mrpDomainLastRingOpenChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Timeticks since last change of ring-state
        to ring open.
    "

    ::= { mrpDomainEntry 16 }

mrpDomainRoundTripDelayMax OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The max. Round-Trip-Delay (in milliseconds)
        which was measured since startup.
    "

    ::= { mrpDomainEntry 17 }

mrpDomainRoundTripDelayMin OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        The min. Round-Trip-Delay (in milliseconds)
        which was measured since startup.
    "

    ::= { mrpDomainEntry 18 }

mrpDomainResetRoundTripDelays OBJECT-TYPE
    SYNTAX      INTEGER {
        resetDelays(1)
    }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "
        A write request with resetDelays(1) shall reset
        the values of mrpDomainRoundTripDelayMax and
        mrpDomainRoundTripDelayMin to zero at the
        same time.
    "

    ::= { mrpDomainEntry 19 }

mrpDomainMRMReactOnLinkChange OBJECT-TYPE
    SYNTAX      INTEGER {
        enabled(1),
        disabled(2)
    }
    MAX-ACCESS   read-create
    STATUS       current
```

```

DESCRIPTION      "
                  Tells whether the MRM reacts on link down MRP link change
                  frames.
enabled(1)        indicates that the MRM reacts immediately on link down MRP
                  link change frames.
disabled(2)       indicates that the MRM does not react on
                  link down MRP link change frames.
                  "
 ::= { mrpDomainEntry 20 }

-- *****
-- additional Objects of the MRP Network Management for management
-- *****

mrpDomainRowStatus OBJECT-TYPE
    SYNTAX          RowStatus
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION     "
                  Support of creation and deletion of rows.
                  A row can be taken out of service by writing
                  the value notInService(2).
                  "
 ::= { mrpDomainEntry 21 }

-- *****
-- Objects of the Media redundancy Interconnection
-- *****

mrpInterconnectionTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF MrpInterconnectionEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "
                  Every entry in this table contains information about
                  a Media redundancy Interconnection domain.

                  Row entries can be created or deleted via
                  SNMP operations. Therefore the MAX-ACCESS values
                  of some Object Identifier are different from the
                  values of the same Object Identifier in the
                  pure monitoring facet.
                  "
 ::= { mrpObjects 2 }

mrpInterconnectionEntry OBJECT-TYPE
    SYNTAX          MrpInterconnectionEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "An entry in the mrpInterconnectionTable."
    INDEX           { mrpInterconnectionID }
 ::= { mrpInterconnectionTable 1 }

MrpInterconnectionEntry ::= SEQUENCE {
    mrpInterconnectionID          Unsigned32,
    mrpInterconnectionName        DisplayString,
    mrpInterconnectionPort        Integer32,
    mrpInterconnectionPortState   INTEGER,
    mrpInterconnectionVlanID      Unsigned32,
    mrpInterconnectionMode         INTEGER,
    mrpInterconnectionAdminRole    INTEGER,
    mrpInterconnectionOperRole     INTEGER,
    mrpInterconnectionState        INTEGER,
    mrpInterconnectionError        BITS,
    mrpInterconnectionOpenCount    Counter32,
    mrpInterconnectionLastOpenChange TimeTicks,
    mrpInterconnectionPeerMAC      OCTET STRING,
    mrpInterconnectionOwnMRMMAC    OCTET STRING,
    mrpInterconnectionPeerMRMMAC   OCTET STRING,
    mrpInterconnectionPeerRingDomainUUID IEC62439UuidType,
    mrpInterconnectionRowStatus    RowStatus
}

mrpInterconnectionID OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "The index of the entry"
 ::= { mrpInterconnectionEntry 1 }

```

```

mrpInterconnectionName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
        A logical name for the Media redundancy Interconnection domain to
        ease the management of Media redundancy Interconnection domains.
    "

 ::= { mrpInterconnectionEntry 2 }

mrpInterconnectionPort OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
        The ifIndex of the layer 2 interface which is used
        as interconnection port.
    "

 ::= { mrpInterconnectionEntry 3 }

mrpInterconnectionPortState OBJECT-TYPE
    SYNTAX      INTEGER {
        disabled(1),
        blocked(2),
        forwarding(3),
        notConnected(4)
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "
        Operational state of the Interconnection port.

        disabled(1) indicates that all frames are dropped.
        blocked(2) indicates that all frames are dropped except
            –MRP Interconnection topology change frames,
            –MRP Interconnection test frames,
            –MRP Interconnection link change frames,
            –MRP Interconnection link status poll frames,
            –frames from other protocols that also define to pass
              blocked(2) ports.
        forwarding(3) indicates that all frames are passed through
            according to the forwarding behavior of IEEE 802.1D.
        notConnected(4) indicates that the port has no link.
    "

 ::= { mrpInterconnectionEntry 4 }

mrpInterconnectionVlanId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
        The VLAN ID assigned to the MRP Interconnection protocol.
        The VLAN ID only is in use when mrpInterconnectionError
        does not signal invalidVlanId.
        If value is set to 0 no VLAN is assigned.
    "

 ::= { mrpInterconnectionEntry 5 }

mrpInterconnectionMode OBJECT-TYPE
    SYNTAX      INTEGER {
        undefined(1),
        lcMode(2),
        rcMode(3)
    }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "
        Control the MRP Interconnection mode.

        The value undefined(1) signals that the interconnection mode is
        neither set to lcMode(2) nor to rcMode(3).
        If the value is set to lcMode(2) the device in Media redundancy
        Interconnection Manager (MI M) role is using the link check mode to
        detect open or closed interconnection links.
        If the value is set to rcMode(3) the device in Media redundancy
        Interconnection Manager (MI M) role is using the ring check mode to
        detect open or closed interconnection links.
    "

```

The factory settings are recommended to adjust the value of this object to the lcMode(2) capability. If the agent restricts the write access, no matter what reason, it shall reject write requests by responding with 'badValue'.

```
::= { mrpInterconnectionEntry 6 }
```

mrpInterconnectionAdminRole OBJECT-TYPE

SYNTAX INTEGER {
disable(0),
client(1),
manager(2)
}

MAX-ACCESS read-write
STATUS current

DESCRIPTION

"Control the MRP Interconnection behavior of the system per Media redundancy Interconnection domain.

If the value is set to disable(0) the MRP Interconnection entity of this domain shall be disabled.
If the value is set to client(1) the entity shall be set to the role of a Media redundancy Interconnection Client (MIC).
If the value is set to manager(2) the entity shall be set to the role of a Media redundancy Interconnection Manager (MIM).
The factory settings are recommended to adjust the value of this object to the client(1) capability of the component in order to prevent multiple managers are in the interconnection (the order of the capabilities are not necessarily conform to the order of the object values here).
If the agent restricts the write access, no matter what reason, it shall reject write requests by responding with 'badValue'.

```
::= { mrpInterconnectionEntry 7 }
```

mrpInterconnectionOperRole OBJECT-TYPE

SYNTAX INTEGER {
disable(0),
client(1),
manager(2)
}

MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The operational role of an MRP Interconnection entity per Media redundancy Interconnection domain.
A value of disabled(0) signals that the entity doesn't work (whatever reason).
A value of client(1) signals that the entity is in a Media redundancy Interconnection Client role.
A value of manager(2) signals that the entity is the manager of this Media redundancy Interconnection domain.

```
::= { mrpInterconnectionEntry 8 }
```

mrpInterconnectionState OBJECT-TYPE

SYNTAX INTEGER {
disable(0),
undefined(1),
interconnOpen(2),
interconnClosed(3)
}

MAX-ACCESS read-only
STATUS current

DESCRIPTION

"Operational status of the MRP Interconnection entity.
disable(0) – The entity is disabled.
undefined(1) – Value is not valid.

interconnOpen(2) – MRP Interconnection redundancy lost of this Media redundancy Interconnection domain.

interconnClosed(3) – MRP Interconnection redundancy available of this Media redundancy Interconnection domain.

```
::= { mrpInterconnectionEntry 9 }
```

```

mrpInterconnectionError OBJECT-TYPE
    SYNTAX      BITS {
                    noError(0),
                    invalidVlanId(1)
                }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                    Configuration status of the MRP Interconnection entity.

                    noError(0)      –no configuration error detected.
                    invalidVlanId(1) –the assigned VLAN ID is not permitted.
                "
 ::= { mrpInterconnectionEntry 10 }

mrpInterconnectionOpenCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                    Counter for MRP Interconnection state changes to
                    interconnOpen.
                "
 ::= { mrpInterconnectionEntry 11 }

mrpInterconnectionLastOpenChange OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                    Timeticks since last change of
                    mrpInterconnectionState to interconnOpen.
                "
 ::= { mrpInterconnectionEntry 12 }

mrpInterconnectionPeerMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                    Device-MAC-Address of the devices interconnection peer, connected
                    to the devices interlink port. This variable contains
                    the MAC-Address contained in the media link check frame received at the
                    interconnection port.
                "
 ::= { mrpInterconnectionEntry 13 }

mrpInterconnectionOwnMRMMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                    The device-MAC-Address of the devices ring manager (MRM).
                    This variable contains the MAC-Address extracted from the
                    periodically received MRP_Test frame sent by the ring manager.
                    If the ring managers MAC-Address is not known, the variable shall
                    contain ff-ff-ff-ff-ff-ff.
                "
 ::= { mrpInterconnectionEntry 14 }

mrpInterconnectionPeerMRMMAC OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                    Device-MAC-Address of devices interconnection peer's, connected
                    to the devices interlink port, ring manager.
                    This variable contains the MRM- MAC-Address of the peers ring manager
                    contained in the media link check frame received at the interconnection port.
                    If the ring managers MAC-Address is not known, the variable shall
                    contain ff-ff-ff-ff-ff-ff.
                "
 ::= { mrpInterconnectionEntry 15 }

mrpInterconnectionPeerRingDomainUUID OBJECT-TYPE
    SYNTAX      IEC62439UuidType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION   "
                    The domain UUID of the ring the interconnection peer's, connected
                    to the devices interlink port, ring.
                    This variable contains the MRM_DomainUUID of the peers ring
                    contained in the media link check frame received at the interconnection port.
                "

```

```

"
::= { mrpInterconnectionEntry 16 }

mrpInterconnectionRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION  "
                Support of creation and deletion of rows.
                A row can be taken out of service by writing
                the value notInService(2).
                "
::= { mrpInterconnectionEntry 18 }

-- *****
-- conformance statements
-- *****
mrpRedundancyGroups OBJECT IDENTIFIER ::= { mrpConformance 1 }

mrpDomainBasicGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainIndex,
        mrpDomainID,
        mrpDomainName,
        mrpDomainAdminRole,
        mrpDomainOperRole,
        mrpDomainRingPort1,
        mrpDomainRingPort1State,
        mrpDomainRingPort2,
        mrpDomainRingPort2State,
        mrpDomainState,
        mrpDomainBlocked,
        mrpDomainRowStatus
    }
    STATUS       current
    DESCRIPTION  "
                Parameters which are necessary to monitor
                and control the status of MRP domains.
                "
::= { mrpRedundancyGroups 1 }

mrpDomainManagerGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainManagerPriority
    }
    STATUS       current
    DESCRIPTION  "
                Parameters which are additional to monitor
                and control the status of MRP domain managers.
                "
::= { mrpRedundancyGroups 2 }

mrpDomainBlockingCntlGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainMRMReactOnLinkChange
    }
    STATUS       current
    DESCRIPTION  "
                Parameters which are optional to monitor and control
                blocking behavior of MRP domains.
                "
::= { mrpRedundancyGroups 3 }

mrpDomainDiagGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainError,
        mrpDomainRingOpenCount,
        mrpDomainLastRingOpenChange,
        mrpDomainRoundTripDelayMax,
        mrpDomainRoundTripDelayMin,
        mrpDomainResetRoundTripDelays
    }
    STATUS       current
    DESCRIPTION  "
                Parameters which are optional to monitor
                and control the diagnostic status of MRP domains.
                "

```

```

::= { mrpRedundancyGroups 4 }

mrpDomainAdvancedGroup OBJECT-GROUP
    OBJECTS {
        mrpDomainVlanId
    }
    STATUS current
    DESCRIPTION
        "
        Parameters which are optional to monitor and control
        the status of MRP domains.
        "

::= { mrpRedundancyGroups 5 }

mrpInterconnectionBasicGroup OBJECT-GROUP
    OBJECTS {
        mrpInterconnectionName,
        mrpInterconnectionPort,
        mrpInterconnectionPortState,
        mrpInterconnectionAdminRole,
        mrpInterconnectionOperRole,
        mrpInterconnectionMode,
        mrpInterconnectionState,
        mrpInterconnectionRowStatus
    }
    STATUS current
    DESCRIPTION
        "
        Parameters which are necessary to monitor
        and control the status of Media redundancy
        Interconnection domains.
        "

::= { mrpRedundancyGroups 6 }

mrpInterconnectionDiagGroup OBJECT-GROUP
    OBJECTS {
        mrpInterconnectionError,
        mrpInterconnectionOpenCount,
        mrpInterconnectionLastOpenChange,
        mrpInterconnectionPeerMAC,
        mrpInterconnectionOwnMRMMAC,
        mrpInterconnectionPeerMRMMAC,
        mrpInterconnectionPeerRingDomainUUID
    }
    STATUS current
    DESCRIPTION
        "
        Parameters which are optional to monitor
        and control the diagnostic status of Media
        redundancy Interconnection domains.
        "

::= { mrpRedundancyGroups 7 }

mrpInterconnectionAdvancedGroup OBJECT-GROUP
    OBJECTS {
        mrpInterconnectionVlanId
    }
    STATUS current
    DESCRIPTION
        "
        Parameters which are optional to monitor and control
        the status of Media redundancy Interconnection domains.
        "

::= { mrpRedundancyGroups 8 }

-- *****
-- compliance specifications
-- *****
mrpRedundancyCompliances OBJECT IDENTIFIER ::= { mrpConformance 2 }

mrpDomainBasicCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "
        Basic implementation requirements for MRP support.
        The agent shall support the monitoring and control
        of redundancy domains.
        "

    MODULE
    MANDATORY-GROUPS {
        mrpDomainBasicGroup
    }

```

```

GROUP      mrpDomainAdvancedGroup
DESCRIPTION
    "
        For SNMP agents it is optional to support
        advanced monitoring and control of
        redundancy domains.
    "
 ::= { mrpRedundancyCompliances 1 }

mrpDomainManagerCompliance MODULE- COMPLIANCE
STATUS      current
DESCRIPTION
    "
        Manager implementation requirements for MRP
        support. The agent shall support the monitoring
        and control of redundancy domains.
    "

MODULE
MANDATORY- GROUPS
    {
        mrpDomainBasicGroup,
        mrpDomainManagerGroup
    }

GROUP      mrpDomainBlockingCtrlGroup
DESCRIPTION
    "
        For SNMP agents it is optional to support blocking control.
    "

GROUP      mrpDomainDiagGroup
DESCRIPTION
    "
        For SNMP agents it is optional to support
        monitoring of the diagnostic status of MRP domains.
    "
 ::= { mrpRedundancyCompliances 2 }

mrpInterconnectionCompliance MODULE- COMPLIANCE
STATUS      current
DESCRIPTION
    "
        Implementation requirements for MRP Interconnection support.
        The agent shall support the monitoring and control
        of Media redundancy Interconnection domains.
    "

MODULE
MANDATORY- GROUPS {
    mrpInterconnectionBasicGroup
}

GROUP      mrpInterconnectionAdvancedGroup
DESCRIPTION
    "
        For SNMP agents it is optional to support
        advanced monitoring and control of Media
        redundancy Interconnection domains.
    "

GROUP      mrpInterconnectionDiagGroup
DESCRIPTION
    "
        For SNMP agents it is optional to support
        monitoring of the diagnostic status of Media
        redundancy Interconnection domains.
    "
 ::= { mrpRedundancyCompliances 3 }

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

```

Annexe A (normative)

Versions antérieures facultatives du protocole du gestionnaire automatique

Un appareil peut prendre en charge la rétrocompatibilité avec les versions antérieures du protocole du gestionnaire automatique. Il doit pour cela mettre en œuvre la fonctionnalité MRA, comme défini dans les Articles 5, 6, 7 et 8, mais aussi:

- Recevoir, analyser et transmettre les cadres MRA MRP_Test avec le contenu des données MRP_Option donné au Tableau A.1:

Tableau A.1 – MRP_Option en mode compatible pour les remplacements MRP_Test

Nom de substitution	Structure
MRP_Option	MRP_TLVHeader (Type=0x7F), MRP_OUI (=0x08-00-06), MRP_SubOption1, [MRP_SubOption2]
MRP_SubOption1	MRP_Ed1Type (=0x00), MRP_Ed1ManufacturerData (=0x00, 0x00)
MRP_SubOption2	MRP_SubTLVHeader (Type=0x03, Longueur=0)

- Recevoir, analyser et transmettre les cadres MRA MRP_TestMgrNAck et MRP_TestPropagate avec le contenu des données MRP_Option donné au Tableau A.2:

Tableau A.2 – Cadres MRP_Option en mode compatible pour les remplacements MRP_TestMgrNAck et MRP_TestPropagate

Nom de substitution	Structure
MRP_Option	MRP_TLVHeader (Type=0x7F, Longueur=0x1A), MRP_OUI (=0x08-00-06), MRP_SubOption1, MRP_SubOption2
MRP_SubOption1	MRP_Ed1Type (=0x00), MRP_Ed1ManufacturerData (=0x00, 0x00)
MRP_SubOption2	MRP_SubTLVHeader (Type=0x01 ou 0x02, Longueur=0x12), MRP_Prio, MRP_SA, MRP_OtherMRMPrio, MRP_OtherMRMSA, Padding (=0x00, 0x00)

Bibliographie

Les documents référencés sont listés dans l'IEC 62439-1. En outre, les documents ci-après sont référencés.

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

IEC 61784-1, *Réseaux de communication industriels – Profils – Partie 1: Profils de bus de terrain*

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

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

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