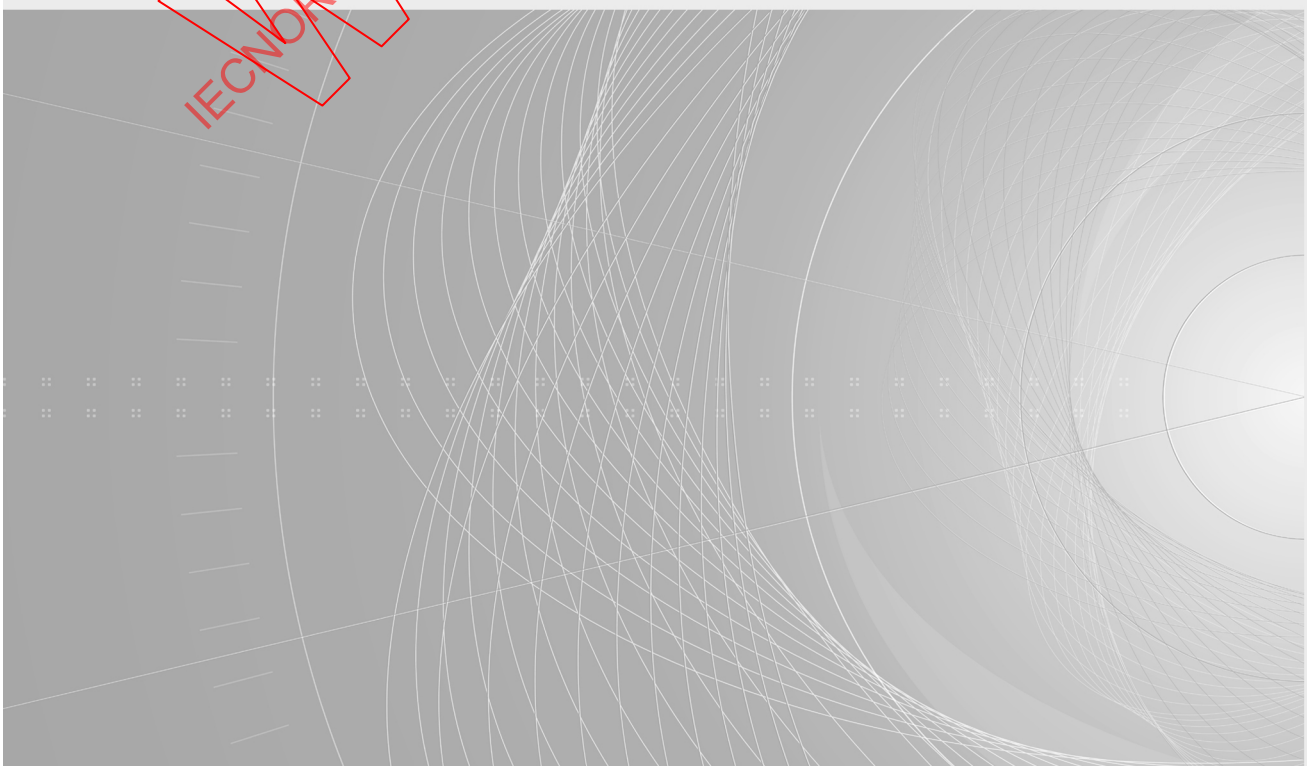
IEC 62439

Edition 1.0 2008-05

INTERNATIONAL STANDARD

High availability automation networks



IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

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

About IEC publications

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

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

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

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC 62439

Edition 1.0 2008-05

INTERNATIONAL STANDARD

High availability automation networks

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XH

ICS 25.040; 35.040

ISBN 2-8318-9765-3

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

HIGH AVAILABILITY AUTOMATION NETWORKS

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 provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of the following patents:

   a) Clause 5 (MRP) may involve Patent WO 99/046908 A1 "Local network, especially Ethernet network, with redundancy properties and redundancy manager", owned by Siemens AG A&D, Gleiwitzerstr. 555, Nürnberg 90475 Germany and Hirschmann Automation and Control GmbH, Stuttgarter Strasse 45-51, Neckartenzlingen 72654, Germany

   b) Clause 6 (PRP) may involve Patent WO06053459 "Reception of redundant and non-redundant frames", owned by ABB Switzerland Ltd, Corporate Research, Segelhofstr 1K, 5405 Baden, Switzerland.

   c) Clause 7 (CRP) may involve Patent U.S. 6,826,590 „Block Oriented Control System on High Speed Ethernet", owned by the Fieldbus Foundation, 9005 Mountain Ridge Drive – Bowie Bldg, Suite190, Austin, TX 78759

   d) Clause 8 (BRP) may involve Patent Application Serial No. US 11/520,192, "Multiple fault-tolerant Ethernet redundancy", owned by Rockwell Automation Technologies, Inc., 1 Allen-Bradley Drive, Mayfield Heights, Ohio, USA

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

   The holders of these patents have assured the IEC that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holders of these patent rights is registered with IEC.

   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."

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

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

FDIS	Report on voting
65C/495/FDIS	65C/498/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.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result 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.

A bilingual version of this publication may be issued at a later date.

INTRODUCTION

This International Standard specifies relevant principles for high availability networks that meet the requirements for industrial automation networks.

In the fault-free state of the network, this International Standard provides ISO/IEC 8802-3 compatible, reliable data communication, and preserves determinism of real-time data communication. In cases of fault, removal, and insertion of a component, it provides deterministic recovery times.

The typical Ethernet communication capabilities as used in the office world are fully retained, 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 Clause 4 and specified in the clauses following it. Clause 4 also distinguishes between the different solutions, giving guidance to the user.

This International Standard follows the general structure and terms of IEC 61158.

HIGH AVAILABILITY AUTOMATION NETWORKS

1 Scope

This International Standard is applicable to high-availability automation networks based on the ISO/IEC 8802-3 (Ethernet) technology.

This International Standard specifies

- a classification scheme for network characteristics (see Annex A);
- a methodology for estimating network availability (see Annex B);
- a set of communication protocols that realize high availability automation networks via the use of redundancy and that can be used in a variety of applications (see Clauses 5, 6, 7, 8).

2 Normative references

The following referenced documents are indispensable for the application of this International Standard. 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 (IEV) – Chapter 191: Dependability and quality of service*

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

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

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

IEC 61918, *Industrial communications networks – Installation of communication networks in industrial premises*

IEEE 802, *IEEE standard for local and metropolitan area networks: Overview and Architecture*

IEEE 802a, *IEEE standard for local and metropolitan area networks: Overview and Architecture*

Amendment 1: *Ethertypes for Prototype and Vendor-Specific Protocol Development*

IEEE 802.1D, *IEEE standard for local and metropolitan area networks: Media Access Control (MAC) bridges*

IEEE 802.1Q, *IEEE standards for local and metropolitan area networks: Virtual bridged local area networks*

IEEE 802.3:2005, *Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

IEEE 1588, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*

DARPA Internet Program Protocol Specification, *Internet Protocol, RFC 791*

3 Terms, definitions, abbreviated terms, 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 the following, apply.

3.1.1
aggregated link
set of inter-switch links configured to work as one inter-switch link

[IEEE 802.3:2005, Clause 43]

3.1.2
aggregated ports
set of inter-switch ports configured to work as one inter-switch port

[IEEE 802.3:2005, Clause 43]

3.1.3
availability (performance)
ability of an item to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval, assuming that the required external resources are provided.

NOTE 1 This ability depends on the combined aspects of the reliability performance, the maintainability performance, and the maintenance support performance.

NOTE 2 Required external resources, other than maintenance resources, do not affect the availability performance of the item.

[IEV 191-02-05]

3.1.4
channel
layer 2 connection between two end nodes which consists of one or more paths (for redundancy) between end nodes

3.1.5
common mode failure
failure that affects all redundant elements for a given function at the same time

3.1.6
complete failure
failure which results in the complete inability of an item to perform all required functions

[IEV 191-04-20]

3.1.7
connection
logical relationship between two nodes

3.1.8
coverage
probability that a failure is discovered within a time short enough for redundancy to handle it, also expressing the percentage of failures caught up by redundancy versus total number of failures

3.1.9
degradation failure
failure which is both a gradual failure and a partial failure

[IEV 191-04-22]

3.1.10
dependability
collective term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance

NOTE Dependability is used only for general descriptions in non-quantitative terms.

[IEV 191-02-03]

3.1.11
device
physical entity connected to the network composed of communication element and possibly other functional elements

[IEC 61158-2, 3.1.12, modified]

NOTE Devices are for instance nodes, routers and switches in this International Standard, an end node and a switch can be combined in one device.

3.1.12
doubly attached node
node that has two ports for the purpose of redundant operation

3.1.13
edge port
port of a switch connected to a leaf link

3.1.14
end node
node which is not able to forward application data to other nodes.

NOTE For the purpose of this standard, further specification is given in 4.2.1.2.

3.1.15
error
discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

NOTE 1 An error can be caused by a faulty item, for example, a computing error made by faulty computer equipment.

NOTE 2 The French term “erreur” may also designate a mistake (see IEV 191-05-25).

[IEV 191-05-24, modified]

3.1.16
extended frame
frame that has been extended by a Redundancy Control Trailer

3.1.17
failure
termination of the ability of an item to perform a required function

NOTE 1 After failure the item has a fault.

NOTE 2 "Failure" is an event, as distinguished from "fault", which is a state.

NOTE 3 This concept as defined does not apply to items consisting of software only.

[IEV 191-04-01]

3.1.18
fault
state of an item characterized by its inability to perform a required function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources

[IEV 191-05-01]

NOTE A fault is often the result of a failure of the item itself, but may exist without prior failure.

3.1.19
fault recovery time
time from the fault event, to the time when the network regains its required communication function in the presence of the fault

NOTE After fault recovery, the network is operating in a degraded mode using some of the redundancy elements, so it has reduced fault resilience, and may not be able to recover from a second fault.

3.1.20
frame
unit of data transmission on an ISO/IEC 8802-3 MAC (Media Access Control) that conveys a protocol data unit (PDU) between MAC service users

[IEEE 802.1Q, modified]

3.1.21
higher-layer entity
entity in a Media Access Control (MAC) Bridge that manages the logical topology of the LAN (spanning tree protocol entity, GARP entity, bridge management, etc.)

[IEEE 802.1D, 7.12.7]

3.1.22
instantaneous failure rate
limit, if it exists, of the quotient of the conditional probability that the instant of a failure of a non-repaired item falls within a given time interval (t, t + Δt) and the duration of this time interval, Δt, when Δt tends to zero, given that the item has not failed up to the beginning of the time interval

[IEV 191-12-02]

NOTE The failure rate is the reciprocal number of the MTTF when the failure rate is constant over the lifetime of one item.

3.1.23
inter-switch link
link between two switches

3.1.24
inter-switch port
port of a switch connected to another switch via an inter-switch link

3.1.25
LAN
part of the network consisting of hubs, switches and inter-switch links, characterized by a common link layer protocol and link address space

NOTE 1 An LAN excludes the end nodes and the leaf links.

NOTE 2 In the context of redundancy, a network may consist of several LANs operated in redundancy.

3.1.26
leaf link
link between an end node and the LAN

NOTE For the purpose of this standard, further specification is given in 4.2.1.3.

3.1.27
linear topology
topology where the switches are connected in series, with two switches each connected to only one other switch and all other switches each connected to two other switches (that is, connected in the shape of a line)

NOTE 1 This topology corresponds to that of an open ring.

NOTE 2 This configuration is sometimes named “daisy chain”. This International Standard does not use the term “daisy chain” because of possible confusion with the term “daisy chain” used elsewhere for busses. From the wiring point of view they require two different implementations.

[IEC 61918, 3.1.39, with modified notes]

3.1.28
link
physical, point-to-point, generally duplex connection between two adjacent nodes

[ISO/IEC 11801, modified]

NOTE “Link” is different from “bus”, which is a broadcast physical medium.

3.1.29
link redundancy entity
entity at layer 2 that hides port redundancy from the upper layers, by forwarding to the upper layers the frames received from the active redundant ports as if they came from a single port, and by forwarding to the active redundant ports a frame coming from the upper layers

3.1.30
link service data unit
data transported within a protocol layer on behalf of the upper layer

NOTE The link service data unit in an Ethernet frame is the content of the frame located between the length/type field and the frame check sequence.

3.1.31
mean failure rate
mean of the instantaneous failure rate over a given time interval $\lambda(t_1, t_2)$.

[IEV 191-12-03]

NOTE This standard uses “failure rate” for the meaning of “mean failure rate” defined by IEV 191-12-03.

3.1.32
mean operating time between failures
expectation of the operating time between failures

[IEV 191-12-09]

3.1.33
mean time to failure (MTTF)
expectation of the time to failure

[IEV 191-12-07]

3.1.34
mean time to recovery (MTTR)
expectation of the time to recovery

[IEV 191-13-08]

3.1.35
mesh topology
topology where each node is connected with three or more inter-switch links

3.1.36
message
ordered series of octets intended to convey information

NOTE Normally used to convey information between peers at the application layer.

[IEC 61784-2, 3.1.14]

3.1.37
network
communication system consisting of end nodes, leaf links and LAN(s)

NOTE A network may have more than one LAN for the purpose of redundancy.

3.1.38
node
network entity connected to one or more links

NOTE Nodes may be either a switch or an end node or both.

[IEC 61784-2, 3.1.16, modified]

3.1.39
partial failure
failure which results in the inability of an item to perform some, but not all, required functions

3.1.40
path
set of links and switches joined in series

NOTE There may be two or more paths between two switches to provide redundancy.

3.1.41
plant
system that depends on the availability of the automation network to operate

EXAMPLE Plants can be power plants, printing machines, manufacturing systems, substations, vehicles.

3.1.42
port
connection point of a node to the network

[ISO/IEC 15802-3]

NOTE 1 This definition is different from a TCP port or a UDP port, which this standard qualifies explicitly, if necessary.

NOTE 2 A port includes layer 1 and 2 implementation.

3.1.43
recovery
event when the network regains the ability to perform its required communication function after a disruption

NOTE Examples of disruptions could be a fault or removal and reinsertion of a component.

3.1.44
recovery time
time period between disruption and recovery

3.1.45
redundancy
existence in an item of two or more means of performing a required function

[IEV 191-15-01]

NOTE In this standard, the existence of more than one path (consisting of links and switches) between end nodes.

3.1.46
reinstatement recovery time
time to reinstate the original, or pre-fault, network configuration, including original operating and management states in each device

3.1.47
reliability
ability of an item to perform a required function under given conditions for a given time interval

[IEV 191-02-06]

NOTE 1 It is generally assumed that the item is in a state to perform this required function at the beginning of the time interval.

NOTE 2 The term "reliability" is also used as a measure of reliability performance (see IEV 191-12-01).

3.1.48
repair
action taken for the re-establishment of the specified condition

3.1.49
repair recovery time
delay between the start of the repair action and the completion of repair of the faulty element such that the network has regained both its required communication function and its required fault resilience

NOTE 1 This time includes any network down time caused by the repair process, for example a network outage to replace a switch with several good ports and one faulty port.

NOTE 2 This time does not include re-instatement time to return the network from its backup mode of operation to the original mode of operation.

3.1.50
ring link
link that connects two switches of a ring

3.1.51
ring port
port of a switch to which a ring link is attached

3.1.52
ring topology
topology in which each node is connected in series to two other nodes

NOTE 1 Nodes are connected to one another in the logical shape of a circle.

NOTE 2 Frames are passed sequentially between active nodes, each node being able to examine or modify the frame before forwarding it.

[IEC 61918, 3.1.57, modified]

3.1.53
robustness
behaviour of the network in face of failures

3.1.54
route
layer 3 communication path between two nodes

3.1.55
single failure criterion
capacity of a system that includes redundant components to maintain its full functionality upon one failure of any of its components, prior to maintenance or automatic recovery

3.1.56
single point of failure
single failure point
component whose failure would result in failure of the system and is not compensated for by redundancy or alternative operational procedure

NOTE A single point of failure or single failure point causes a common mode failure. It may be caused by a design error in the redundant elements or by an external cause that affects all redundant elements in the same way, for example, extreme temperature.

3.1.57
singly attached node
node that has only one port to a LAN

3.1.58
stand-by redundancy
redundancy wherein a part of the means for performing a required function is intended to operate, while the remaining part(s) of the means are inoperative until needed

[IEV 19-5-03]

NOTE This is also known as dynamic redundancy.

3.1.59
star topology
topology in which all devices are connected to a central node

[IEC 61918, 3.1.63, modified]

3.1.60
switch
switch node
MAC bridge as defined in IEEE 802.1D

NOTE The term “switch” is used as a synonym for the term “switch node”.

3.1.61
systematic failure
failure related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors

NOTE 1 Corrective maintenance without modification usually eliminates the failure cause.

NOTE 2 A systematic failure can be induced by simulating the failure cause.

[IEV 191-04-19]

3.1.62
topology
pattern of the relative positions and interconnections of the individual nodes of the network

[IEC 61918, 3.1.67, modified]

NOTE Additional aspects such as the delay, attenuation and physical media classes of the paths connecting network nodes are sometimes also considered to be properties of the topology.

3.1.63
tree topology
topology in which any two nodes have only one path between them and at least one switch is attached to more than two inter-switch links

3.1.64
trunk portion
part of a switched LAN that carries traffic for several end nodes

3.1.65
unavailability
state of an item of being unable to perform its required function

[IEV 603-05-05]

NOTE Unavailability is expressed as the fraction of expected operating life that an item is not available, for example given in minutes per year.

3.1.66
upper layer entity
parts of the protocol stack immediately above the redundancy handling layer

[IEV 603-05-05]

3.1.67
worst case recovery time
maximum expected recovery time amongst all faults and for all allowed configurations

NOTE This delay is important for a network designer to indicate which aspects of the network need special treatment to minimize communication disruption.

3.2 Abbreviated terms and acronyms

AL Application Layer

ARP Address Resolution Protocol

ASE Application Service Element

BPDU Bridge management Protocol Data Unit, according to IEEE 802.1D

BRP	Beacon Redundancy Protocol, according to Clause 8.
CRP	Cross-network Redundancy Protocol, according to Clause 7
DAN	Doubly attached node
DANB	Double attached node implementing BRP, according to Clause 8
DANC	Doubly attached node implementing CRP, according to Clause 7
DANP	Double attached node implementing PRP, according to Clause 6
DLPDU	Data Link Protocol Data Unit
DLSDU	Data Link Service Data Unit
FCS	Frame Check Sequence, IEEE 802.3 cyclic redundancy check
FDB	Filtering Data Base
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Network
LLDP	Link Layer Discovery Protocol, see IEEE 802.1AB:2005
LRE	Link Redundancy Entity
LSB	Least Significant Bit
LSDU	Link Service Data Unit
MAC	Media Access Control
MIB	Management Information Base
MRC	Media Redundancy Client, see Clause 5
MRM	Media Redundancy Manager, see Clause 5
MRP	Media Redundancy Protocol, see Clause 5
MTBF	Mean Time Between Failure
MTTF	Mean Time To Failure
MTTFN	Mean Time To Failure of Network
MTTFS	Mean Time To Failure of System
MTTR	Mean Time To Repair
MTTRP	Mean Time To Repair Plant
OUI	Organizational Unique Identifier
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
PRP	Parallel Redundancy Protocol, see Clause 6
PTP	Precision Time Protocol, see IEEE 1588

QAN Quadruply Attached Node

RCT Redundancy Control Trailer, see Clause 6

RSTP Rapid Spanning Tree Protocol, see IEEE 802.1D:2004

SAN Singly Attached Node

SANC Singly Attached Node implementing CRP, see Clause 7

SNMP Simple Network Management Protocol

SRP Serial Redundancy Protocol, see Clause 6

VDAN Virtual Doubly Attached Node, see Clause 6

VLAN Virtual LAN, see IEEE 802.1Q

3.3 Conventions

3.3.1 General conventions

The protocols specified in this standard follow the structure defined in IEC 61158-1.

General guidelines are specified in IEC 61158-6-10, 3.6.

3.3.2 Conventions for state machine definitions

This standard follows the conventions used in IEC 61158-6-10, 3.7. The following is a summary.

- Each state is described by one table, with a separate row for each transition that may cause a state change.
- Transitions are defined as events that may carry arguments and be subject to conditions.
- The action field expresses the action that take place in case the event is fired.
- For space reasons, the event and the actions are in the same cell.
- The right column indicates the next state that is entered after the action is finished.

3.3.3 Conventions for PDU specification

PDUs are described according to specification DARPA RFC 791, Appendix B.

In particular:

- bits, octets and arrays are numbered starting with 0;
- the Network Byte Ordering (big-endian, most significant octet first) convention is observed.

IEC 61158-6-10 distinguishes bit identification from the bit offset.

EXAMPLE In a bit string of 8 bits, the rightmost bit (LSB) is labelled bit 0, but it has bit offset 7 within the bit string octet.

When specifying data objects rather than PDUs, the bit identification according to IEC 61158-6 is used. Consequently, bits of a bit string are specified in ascending bit identification, although they are transmitted in the opposite order.

3.4 Reserved network addresses

The following is a summary of the network addresses reserved for the purpose of this standard, whilst the prescribed values are specified in the respective clauses.

For the purpose of this standard, the OUI 01-15-4E has been reserved by IEEE. All bands within this OUI are reserved for this standard. The following bands are assigned:

- MRP (see Clause 5) uses 01-15-4E, band 00-00-xx.
- PRP (see Clause 6) uses 01-15-4E, band 00-01-xx.
- CRP (see Clause 7) uses an IP multicast MAC address.
- BRP (see Clause 8) uses 01-15-4E, band 00-02-xx.

For the purpose of this standard, the following Ethertypes (see IEEE 802a) have been reserved by IEEE:

- MRP (see Clause 5) uses 0x88E3.
- PRP (see Clause 6) uses 0x88FB.
- CRP (see Clause 7) uses 0x0800 (IP) with UDP port 3622.
- BRP (see Clause 8) uses 0x80E1.

4 Concepts for high availability automation networks

4.1 Characteristics of application of automation networks

4.1.1 Resilience in case of failure

Plants rely on the correct function of the automation system. Plants tolerate a degradation of the automation system for only a short time, called the grace time. The network recovery time should be shorter than the grace time since the application typically needs to perform additional tasks (related to protocol and data handling, waiting for the next scheduled communication cycle, etc.) before the plant is back to the fully operational state. Applications can be distinguished by their grace time, as Table 1 shows.

Table 1 – Examples of application grace time

Applications	Typical grace time
Uncritical Automation, for example, enterprise systems	20 s
Automation management, for example, manufacturing, discrete automation	2 s
General automation, for example, process automation, power plants	0,2 s
Time-critical automation, for example, synchronized drives	0,020 s

Some plants have stricter requirements when they are required to operate continuously, having no idle period during which the plant may be maintained or reconfigured. In this case, the grace time holds for the stricter requirement, for instance, dictated by the hot-swapping of parts of the equipment.

Automation systems may contain redundancy to cope with failures. Methods differ on how to handle redundancy, but their key performance factor is the recovery time, i.e., the time needed to restore operation after occurrence of a disruption. If the recovery time exceeds the grace time of the plant, protection mechanisms initiate a (safe) shutdown, which may cause significant loss of production and plant operational availability.

A key characteristic of recovery is its determinism, i.e., the guarantee that the recovery time remains below a certain value as long as the basic assumptions (single failure at a time, no common mode of failure, less than maximum system extension) are met.

Whenever operation depends on the correct function of the automation network, it may become necessary to increase the availability of the network.

Raising availability by increasing reliability of the elements or improving maintenance is outside the scope of this standard. This standard considers only protocols that introduce redundancy and automatically reconfigure redundant network elements in case of failure.

4.1.2 Classes of network redundancy

4.1.2.1 General

This standard considers two classes of network redundancy:

- redundancy managed within the network;
- redundancy managed in the end nodes.

NOTE This standard does not consider redundancy of the end nodes themselves, i.e., the use of redundant end nodes, since this is highly application specific.

4.1.2.2 Redundancy managed within the network

Redundancy within a network has been applied to wide area networks and to legacy field busses.

Layer 3 routers (not considered in this standard) calculate alternate routes upon link failures. The corresponding protocols are well proven as part of the IP suite, but the recovery time is in the order of dozen of seconds, if not minutes, depending on the topology. Such recovery times are only tolerated by the most benign applications.

Automation networks usually operate within one single Local Area Network (LAN), i.e., messages for operation are threaded through layer 1 repeaters or layer 2 switches, but do not cross routers. Messages to and from the outside world over routers or firewalls do exist, but are considered to be uncritical.

Classically, redundancy within a LAN is handled by protocols that react to loss of links and switches by reconfiguring the LAN, using redundant links and switches, such as the Rapid Spanning Tree Protocol (RSTP) according to IEEE 802.1D.

Improved Layer 2 redundancy protocols build on similar principles as RSTP, but provide a faster recovery by exploiting the assumption that the automation network has a ring topology. End nodes are unmodified automation nodes.

4.1.2.3 Redundancy managed in the end nodes

Further improvements in recovery time require managing of redundancy in the end nodes, by equipping the end nodes with several, redundant communication links. In general, doubly attached end nodes provide sufficient redundancy. In this type of redundancy, no assumption about the switches within the LAN is made.

For time-critical applications such as synchronized drives, the parallel operation of disjoint networks provides a seamless recovery, but requires complete duplication of the network. Some critical plants also require doubly attached nodes in order to cope with a failure of a leaf link, even if they do not require a very short recovery time.

4.1.3 Redundancy maintenance

Redundancy can be affected by latent faults, which can be detected by testing. The testing interval allows availability to be estimated. All protocols provide the means to test the redundant or spare components and report detected failures to the network management.

4.1.4 Comparison and indicators

The protocols specified in this standard offer

- a maximum, deterministic and guaranteed recovery time (that may depend on the topology);
- transparency of the actual communication towards the application under all circumstances; and
- for doubly attached nodes, interoperability with singly attached devices (off-the-shelf, IT equipment).

Table 2 compares some characteristics of some redundancy protocols, ordered by recovery time.

Table 2 – Examples of redundancy protocols

Protocol	Solution	Frame loss	Redundancy protocol	End node attachment	Network topology	Recovery time fault or repair
IP	IP routing	Yes	Within the network	Single	Single meshed	>30 s typical not deterministic
STP	IEEE 802.1D:1998	Yes	Within the network	Single	Single meshed	>20 s typical not deterministic
RSTP	IEEE 802.1D:2004	Yes	Within the network	Single	Single meshed	>2 s typical not deterministic
CRP	IEC 62439, Clause 7	Yes	In the end nodes	Single and double	Connected, doubly meshed	1 s worst case for 512 end nodes
MRP	IEC 62439, Clause 5	Yes	Within the network	Single	Ring	200 ms worst case for 50 switches
BRP	IEC 62439, Clause 8	Yes	In the end nodes	Double	Connected, doubly meshed	4,8 ms worst case for 500 end nodes
PRP	IEC 62439, Clause 6	No	In the end nodes	Double	Independent double meshed	0 s

NOTE For the redundancy protocols specified in this standard, the recovery times in Table 2 are guaranteed when using the specified settings and parameters. Faster recovery times may be achieved using different settings and parameters under the user's responsibility.

The indicators for the different solutions include, when applicable,

- fault recovery time;
- repair recovery time;
- reinstatement recovery time;
- worst-case recovery time;
- impact on normal operation.

The fault cases include

- failure of the current active network manager (if it exists) followed by repair and reinstatement;
- failure of the current source of network time (if it exists), followed by repair and reinstatement.

Subclause 4.2 generalizes the above considerations and introduces a classification scheme.

4.2 Generic network system

4.2.1 Network elements

4.2.1.1 General

The generic network is modelled with the functional elements listed below and represented in Figure 1.

- End nodes
- Leaf links
- Switches (with edge ports and inter-switch ports)
- Inter-switch links
- Switching end nodes

The LAN consists of all network components, except the end nodes and leaf links.

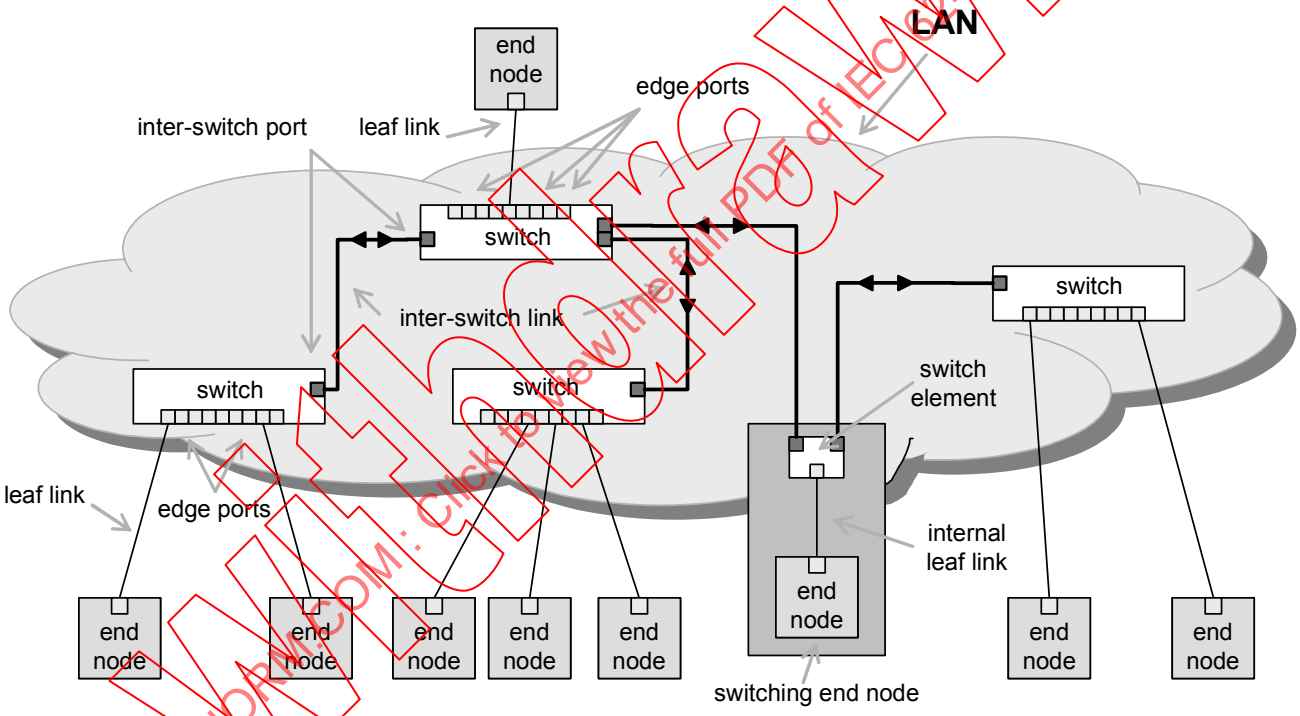


Figure 1 – General network elements (tree topology)

NOTE Edge ports are shaded in light grey, inter-switch ports are shaded in dark grey, inter-switch links are drawn with a thick line, leaf links drawn with a thin line.

4.2.1.2 End node

An end node requires one connection port to the LAN for its normal operation.

The connection port of an end node is connected to an edge port of a switch in a LAN by a leaf link.

4.2.1.3 Leaf link

A leaf link connects an end node with a LAN.

This connection may be internal to a device, in the case where the device combines the end node and switch or LRE functionality (switching end node in Figure 1).

4.2.1.4 Inter-switch link

An inter-switch link connects the switches within a LAN.

There may be several inter-switch links between two switches to increase availability.

4.2.1.5 Switches

Switches are layer 2 connecting elements as defined in IEEE 802.1D.

NOTE Bridges according to IEEE 802.1D are called switches in this standard.

Switches are connected to each other by inter-switch links.

A switch is connected to a leaf link through an edge port.

4.2.1.6 Switching end node

A switch element may be implemented within the same piece of physical equipment as the end node. Although this makes the end node appear to be a doubly attached node, internally the operating principle is different, since there is no need for a link redundancy entity (LRE) because the switch element plays this role.

4.2.1.7 End nodes with multiple attachments

End nodes may have more than one connection port for redundancy. Connection ports of an end node may be connected to the same LAN or may be connected to different LANs.

End nodes with more than one attachment require an LRE in their communication stack to hide redundancy from the application (see Figure 22 or Figure 31 or Figure 39).

An end node connected to one or two LANs of the same network through two leaf links is a doubly attached node (DAN).

An end node connected to one or more LANs of the same network through four leaf links is a quadruply attached node (QAN).

NOTE End nodes using different communication ports for independent networks are not considered here, the considerations apply to each network separately.

4.2.2 Topologies

4.2.2.1 General

Redundancy within the network considers the presence of more network elements (switches, links) than necessary for operation, in order to prevent loss of communication caused by a failure. To this effect, there is more than one physical path between any two end nodes.

IEC 61918 specifies various kinds of basic physical topologies, some of which are used by this standard to define different topologies.

a) Topologies without redundancy
   - Tree topology (Figure 2)
   - Linear topology (Figure 3)

b) Topologies with redundant links
   - Ring topology (Figure 4)
   - Partial meshed topology (Figure 5)

– Fully meshed topology (Figure 6)

There are four top level structures.

- Single LAN without redundant leaf links (see 4.2.2.4.1)
- Single LAN with redundant leaf links (see 4.2.2.4.2)
- Redundant LANs without redundant leaf links (see 4.2.2.4.3)
- Redundant LANs with redundant leaf links (see 4.2.2.4.4)

When redundancy is handled in the LAN, end nodes can be singly attached. In the case of switch or leaf link failure, such end nodes may lose communication.

4.2.2.2 Topologies without redundancy

4.2.2.2.1 Tree topology

In a tree topology, at least one switch has more than two inter-switch links and there is only one path between any two devices. Figure 2 shows an example of tree topology.

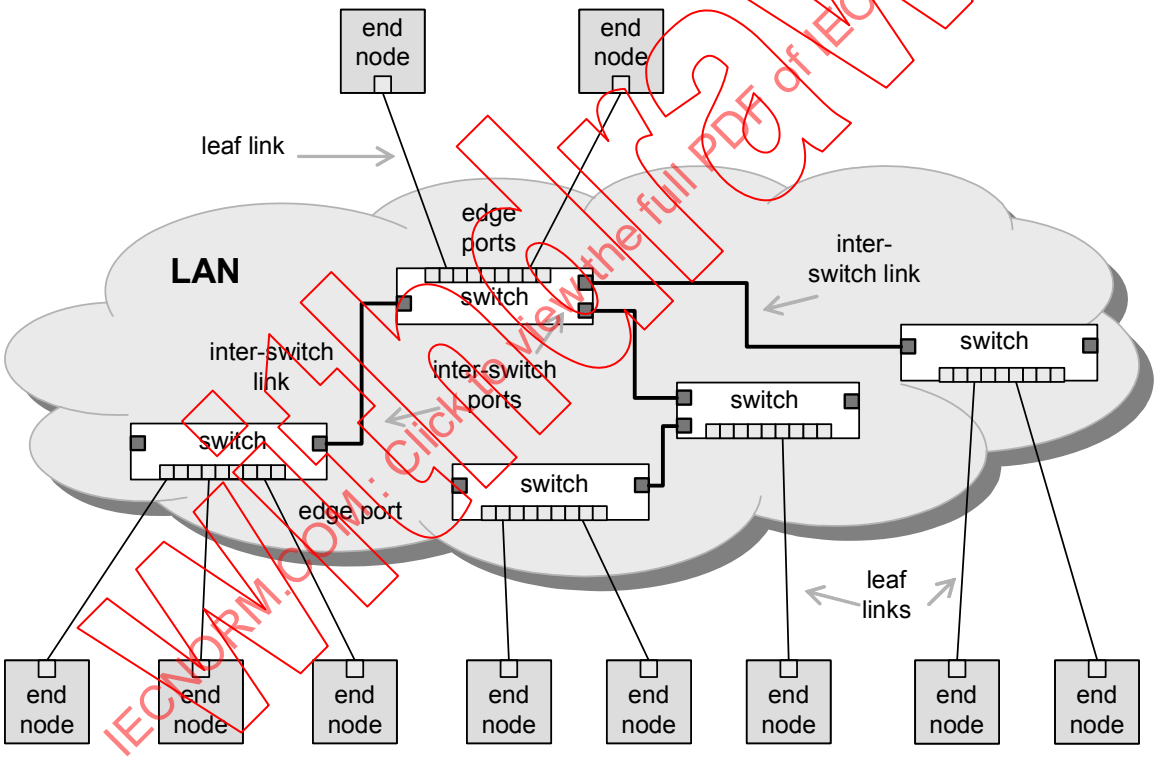


Figure 2 – Example of tree topology

4.2.2.2.2 Linear topology

In a linear topology, all switches are connected to each other in line and no node has more than two inter-switch links but the two nodes located at the end of the line have only one inter-switch link. Figure 3 shows an example of linear topology.

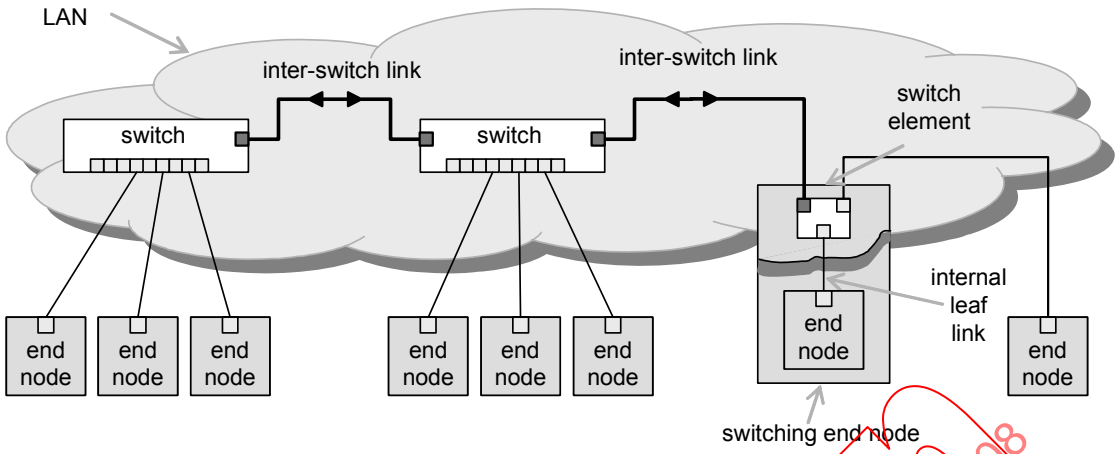


Figure 3 – Example of linear topology

NOTE A node may be a switching end node, as shown in the second rightmost end node of Figure 3.

4.2.2.3 Topologies with redundant links

4.2.2.3.1 Ring topology

NOTE This topology applies to MRP redundancy (see Clause 5).

In a ring topology, every switch has two inter-switch links and any two end nodes have two paths between them when all components are operational. Figure 4 shows an example for the ring topology.

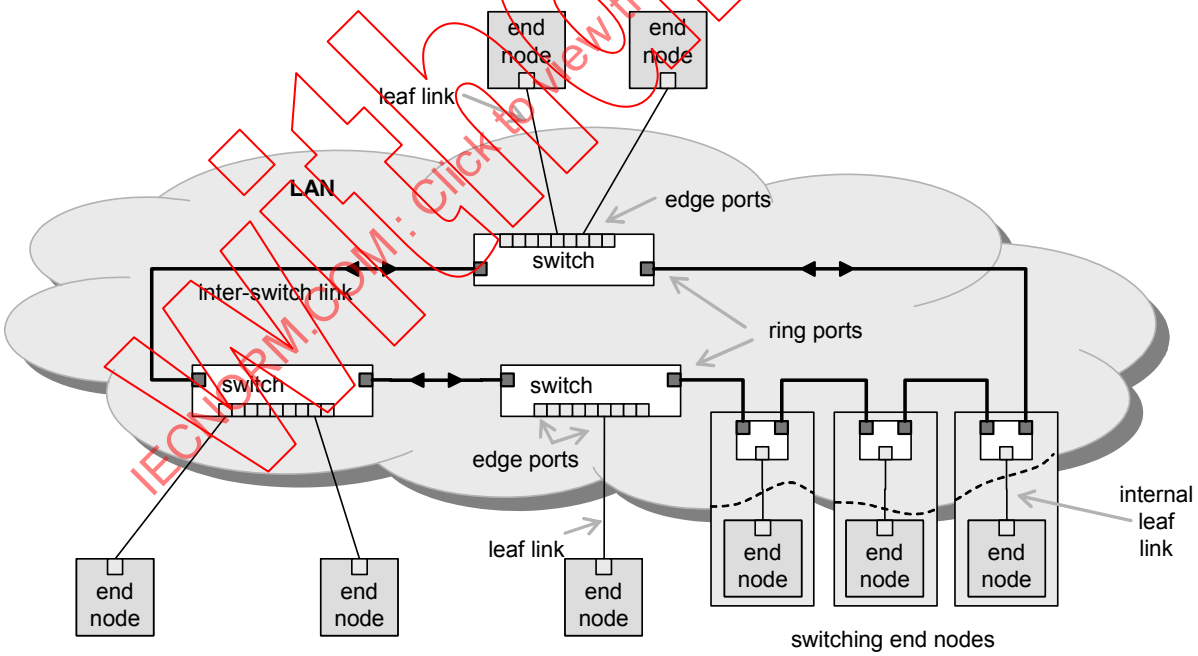


Figure 4 – Example of ring topology

A ring topology introduces a loop in the LAN that could lead to flooding by permanently circulating frames. Protocols such as the rapid spanning tree protocol (RSTP) and the media redundancy protocol (MRP) ensure that the switches maintain a logical linear topology during initialization, operation and reconfiguration.

If a switch or an inter-switch link fails, the switch is excluded from the ring, and a new logical linear topology is established. However, end nodes connected to a failed switch lose connectivity.

4.2.2.3.2 Partially meshed topology

In a partially meshed topology, at least one switch has more than two inter-switch links and there exist more than one path between some devices. Figure 5 shows an example of a meshed topology.

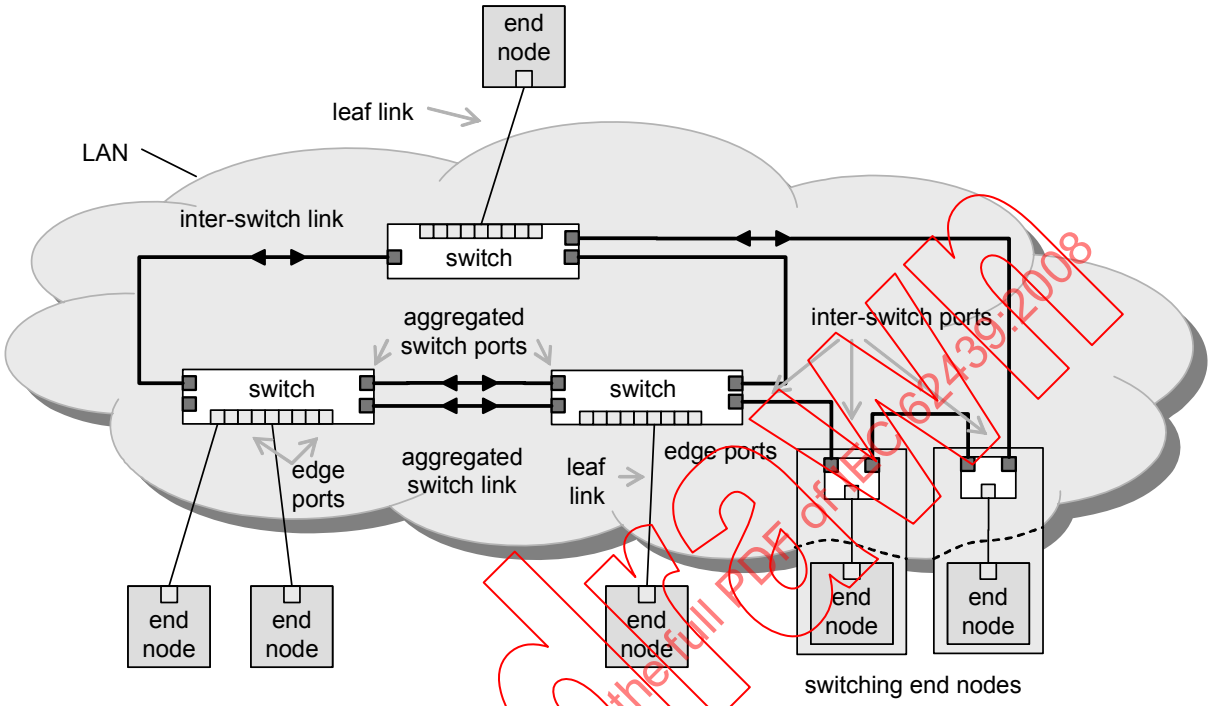


Figure 5 – Example of a partially meshed topology

4.2.2.3.3 Fully meshed topology

In a fully meshed topology, every switch has more than two inter-switch links.

In a fully meshed topology, the failure of any inter-switch link and of any switch can be tolerated. However, end nodes connected to a failed switch lose connectivity. Figure 6 shows an example of a fully meshed topology.

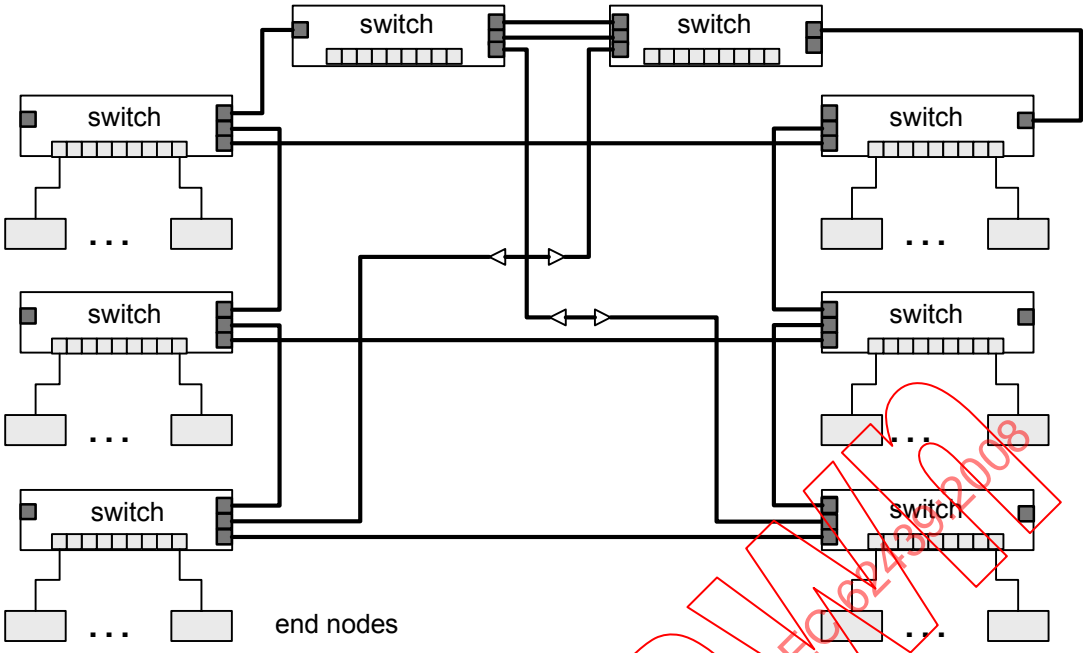


Figure 6 – Example of fully meshed topology

4.2.2.4 Top level structures of networks

4.2.2.4.1 Single LAN without redundant leaf links

This topology has only one path between any two nodes (see Figure 7).

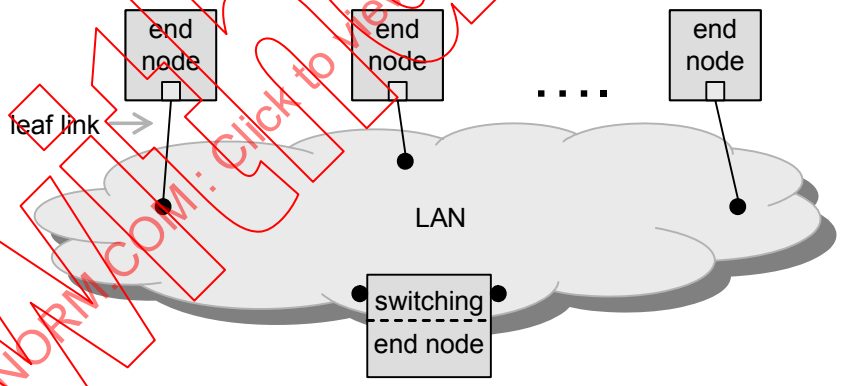


Figure 7 – Single LAN structure without redundant leaf links

Examples of this topology are the tree and linear topologies (see Figure 2 and Figure 3).

4.2.2.4.2 Single LAN with redundant leaves

NOTE This topology applies to SRP (see Clause 6).

DANs are connected to the same LAN through leaf links. Each edge port may belong to the same switch or to different switches. Figure 8 gives an example.

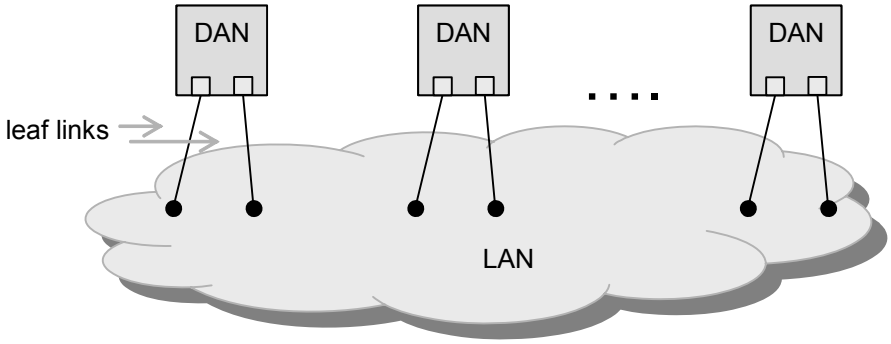


Figure 8 – Single LAN structure with redundant leaf links

4.2.2.4.3 Network without redundant leaves

NOTE This topology applies to PRP (see Clause 6), CRP (see Clause 7) and BRP (see Clause 8).

In this type of topology, paths do not overlap. Redundant leaf links are connected to different LANs. An example is shown in Figure 9.

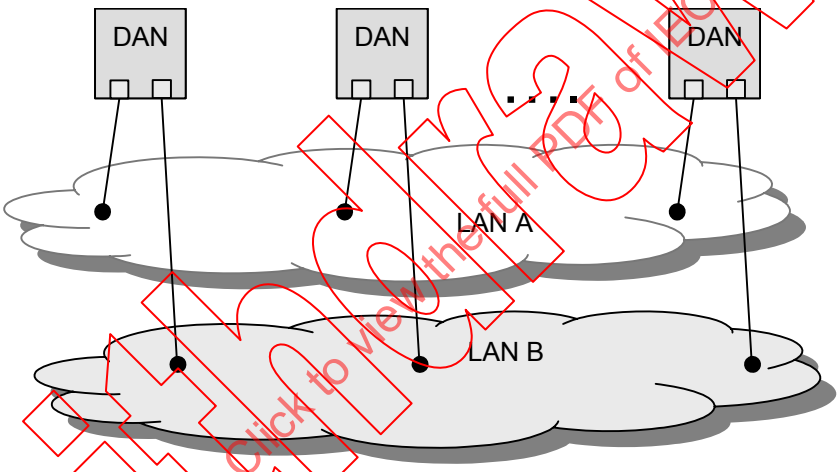


Figure 9 – Redundant LAN structure without redundant leaf links

4.2.2.4.4 Network with redundant leaf links

Redundant leaf links are connected both to the same LAN and different LANs. Nodes are quadruply attached nodes (QANs). An example is shown in Figure 10.

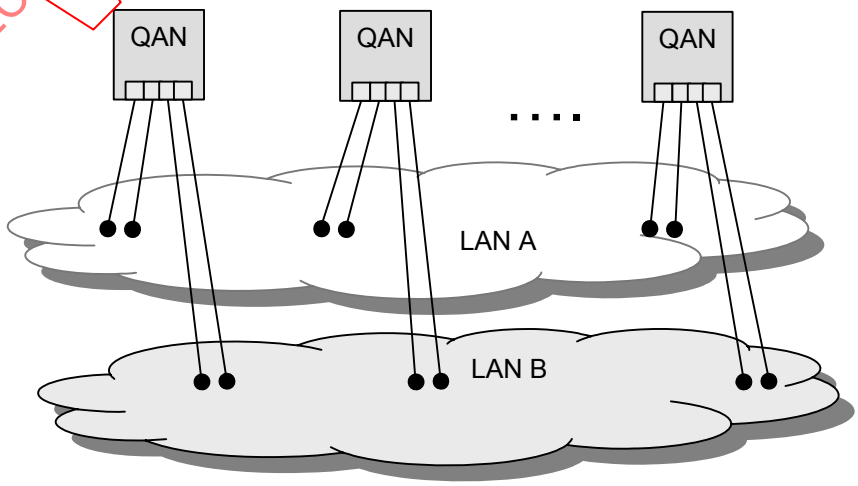


Figure 10 – Redundant LAN structure with redundant leaf links

4.2.3 Redundancy handling

4.2.3.1 Backup mode

In the backup mode, only one of the redundant paths is selected as on-service while the other paths are in stand-by.

If the on-service path becomes unavailable, another path backs it up.

During the elapsed time from the loss of the on-service path to the beginning of operation of the backup path, messages can be lost, therefore the channel is considered in disconnected state.

NOTE IEV calls this kind of redundancy “stand-by” or “passive” redundancy. The term “dynamic redundancy” is also used.

4.2.3.2 Alternate (active) mode

In the alternate mode, redundant paths are used alternately, at random or according to regular patterns, and messages are transmitted via one of the redundant paths.

If it is detected that one of the redundant paths is in disconnected state, that path stops being used while other paths continue being used alternatively.

This mode allows checking the availability of the components continuously and therefore increases coverage.

4.2.3.3 Parallel (active) operation

In the parallel operation, messages are transmitted via all available redundant paths.

The receiving end node selects one of the received messages.

NOTE The term “static redundancy” or “work-by” is also used.

4.2.4 Network recovery time

Network recovery time is called recovery time in this standard because this standard deals only with networks. The definition in 3.1.44 applies.

4.2.5 Diagnosis coverage

Faults are detected through error-detection mechanisms that detect only a percentage of the faults. The coverage is the probability that diagnosis mechanisms detect an error within a time that allows recovery before other mechanisms take action to protect the plant or before the plant suffers damage.

4.2.6 Failures

4.2.6.1 Kinds of failure

There are three kinds of failure:

- transient failure;
- component failure; and
- systematic failure.

They affect the following elements:

- end nodes;

- leaf links;
- switches;
- inter-switch links.

4.2.6.2 Transient failures

A transient failure such as EM interferences causes transient errors, which leave the hardware essentially intact but disrupt the function. In this case, the failed part can be automatically reintegrated after automatic testing. Such mechanisms are partially implemented in the redundancy protocols specified in this standard.

NOTE EM interferences can become systematic failures.

4.2.6.3 Component failure

A component failure may be partial or complete. Only complete failures of components (not intermittent, not spurious) are considered in this standard.

4.2.6.4 Systematic failure

A systematic failure affects several redundant components at the same time; it is therefore a single point of failure. Configuration errors also belong to this category. The redundancy protocols specified in this standard do not consider systematic failures but allow detecting some.

NOTE Diversity of the design is possibly able to reduce impact of systematic failure.

4.2.6.5 End node failure

End node failure is outside the scope of this standard.

4.2.6.6 Leaf link failure

Leaf link failure is caused by

- failure of the connection port of end node;
- failure of the leaf link cable; or
- failure of the edge port.

4.2.6.7 Switch failure

A switch consists of a core switch functionality (for instance, processor, power supply) and a number of ports.

For calculation purposes, a switch failure considers only the failure of the core switch function.

Failure of an edge port of the switch is considered as a leaf link failure.

Failure of an inter-switch port of the switch is considered as an inter-switch link failure.

4.2.6.8 Inter-switch link failure

Inter-switch link failure is caused by

- failure of either inter-switch port; or
- failure of the inter-switch link cable.

4.3 Safety

This standard does not consider safety aspects such as integrity.

NOTE Even though safety is not directly addressed, high reliability is a desirable feature in a safety system.

4.4 Security

This standard does not consider security (for example privacy, authentication) issues.

4.5 Conformance

4.5.1 Conformance to redundancy protocols

A statement of compliance with a clause of this standard shall be stated as:

- compliance to IEC 62439, Clause 5 (MRP); or
- compliance to IEC 62439, Clause 6 (PRP); or
- compliance to IEC 62439, Clause 7 (CRP); or
- compliance to IEC 62439, Clause 8 (BRP).

A conformance statement shall be supported with appropriate documentation as defined in 4.5.2. The supported protocols and options shall be specified as PICS, in the format PICS_62439-X_supported options.

Example PICS_62439-5_BlockingSupported.

4.5.2 Conformance tests

4.5.2.1 Concept

The concept of this conformance test is to verify the capabilities of a device under test (DUT) against a consistent set of indicators under simulated worst-case conditions. The conformance test shall assert the interoperability of devices which claim compliance with the same protocol.

This standard contains specifications that are to be observed by the following different actors:

- the device builder, who designs and tests a compliant interface;
- the network manager, who defines the topology;
- the user of the network, who respects the operational limitations.

A device sold as being fully compliant with this standard could underperform if the network configuration rules are not observed when it is used.

Figure 11 gives an overview of the conformance test related to this standard.

NOTE Conformance test implementation and conformance test execution are not defined in this standard.

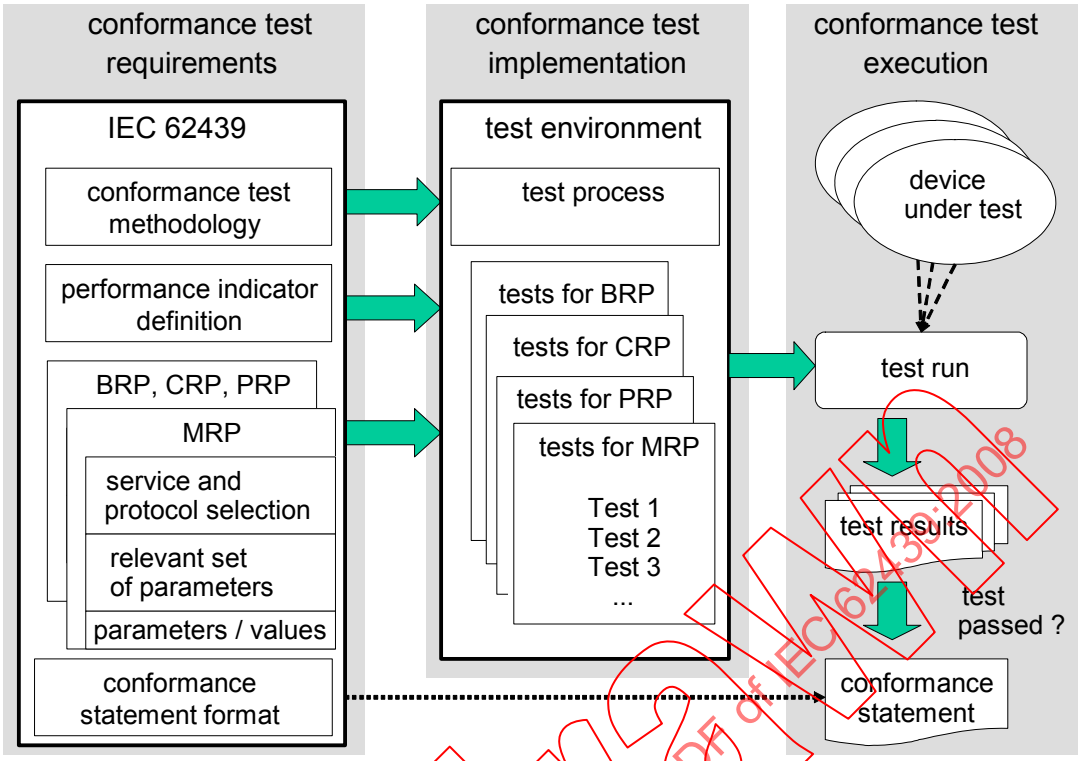


Figure 11 – Conformance test overview

4.5.2.2 Methodology

Test cases shall be developed in a way that tests are repeatable. Test results shall be documented and shall be used as the basis for the conformance statement.

Conformance tests of a device shall include, as appropriate, the verification of

- correctness of the specified functionality;
- network-related indicator values;
- device-related indicator values.

The performance indicator values of the protocol and of the device under test shall be used.

NOTE 1 A description of a conformance testing process is given in ISO/IEC 14496-4.

NOTE 2 It is assumed that the quality of the test cases guarantees the interoperability of a tested device. If any irregularities are reported, the test cases will be adapted accordingly.

4.5.2.3 Test conditions and test cases

Test conditions and test cases shall be defined and documented based on a specific redundancy protocol. This shall include the following indicators, when applicable:

- number of nodes;
- network topology;
- number of switches between nodes;
- type of traffic.

For each measured indicator, test condition and test case documents shall be prepared and shall describe

- test purpose;

- test set-up;
- test procedure;
- criteria for compliance.

Test set-up describes the equipment set-up necessary to perform the test including measurement equipment, DUT, auxiliary equipment, interconnection diagram, and test environmental conditions.

Parts of the test environment may be emulated or simulated. The effects of the emulation or simulation shall be documented.

The test procedure describes how the test should be performed, which also includes a description of a specific set of indicators required to perform this test. The criteria for compliance define test results accepted as compliance with this test.

4.5.2.4 Test procedure and measuring

The measured indicators shall include, when applicable,

- redundancy recovery time;
- impact of redundancy overhead on normal operation.

The test procedure shall be based on the principles of 4.5.2.3.

The sequence of measuring actions to complete a test run shall be provided.

The number of independent runs of the test shall be provided.

The method to compute the result of the test from the independent runs shall be provided if applicable.

4.5.2.5 Test report

The test report shall contain sufficient information so that the test can be repeated.

The test report shall contain at least

a) the reference to the conformance test methodology according to 4.5.2.2;
b) the reference to the performance indicator definitions;
c) the reference to the used redundancy protocol according to this standard,
d) a description of the conformance test environment including network emulators, measurement equipment and the person or organization responsible for the test execution, and the date of testing;
e) a description of the DUT, its manufacturer, and hardware and software revision;
f) the number and type of devices connected to the network together with the topology;
g) a reference to the test case specifications;
h) the measured values;
i) a statement regarding compliance with the redundancy protocol.

5 MRP – Media Redundancy Protocol based on a ring topology

5.1 MRP Overview

The Media Redundancy Protocol (MRP) specifies a recovery protocol based on a ring topology.

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

MRP is based on functions of ISO/IEC 8802-3 (IEEE 802.3) and IEEE 802.1D including the filtering data base (FDB) and is located between the data link layer and application layer (see Figure 12).

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

A 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.

A compliant node shall have the ability to perform as one of the following:

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

Each MRP-compliant node requires a switch element with two ring ports connected to the ring.

NOTE Additional ring ports may be used to connect to another ring.

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 neighbouring node (see 5.2.1).

The MRP consists of a service and a protocol entity, see stack model in Figure 12.

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.

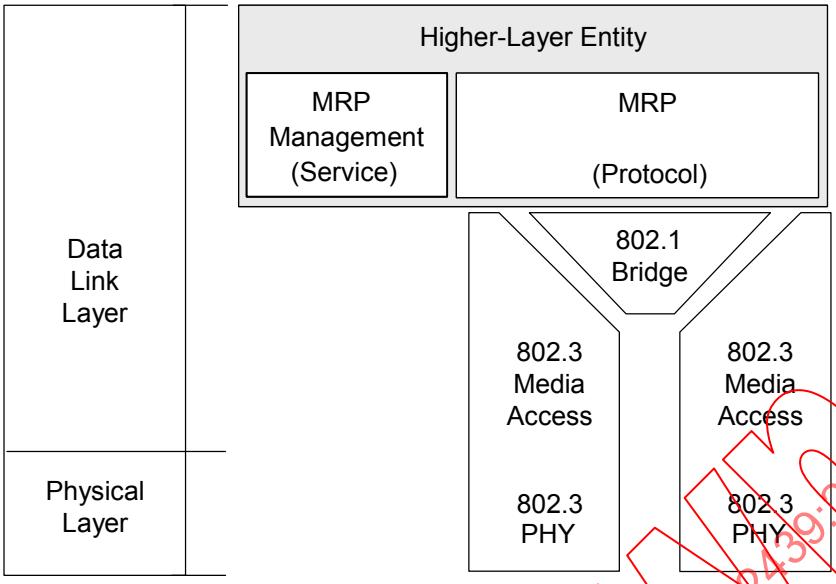


Figure 12 – MRP stack

5.2 MRP Media redundancy behaviour

5.2.1 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 IEEE 802.3.

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

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

- DISABLED:

  All frames shall be dropped.
- BLOCKED:

  All frames shall be dropped except:
  - MRP_TopologyChange frames and MRP_Test frames from the MRM.
  - MRP_LinkChange frames from an MRC.
  - frames from other protocols that are defined in IEEE 802.1D, Tables 7-10 to pass ports that are in BLOCKED state (for example, LLDP, PTP).
  - frames with a destination address equal to a group address configured in the Permanent Database (see IEEE 802.1D, 7.12.6) marked to pass ports that are in BLOCKED state.
- FORWARDING:

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

NOTE 1 Designers should be aware that switches in a ring should not block additional services which are supported by the switches.

NOTE 2 IEEE 802.1D calls the port state corresponding to BLOCKED as BLOCKING.

5.2.2 Media Redundancy Manager (MRM)

The first 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 13.

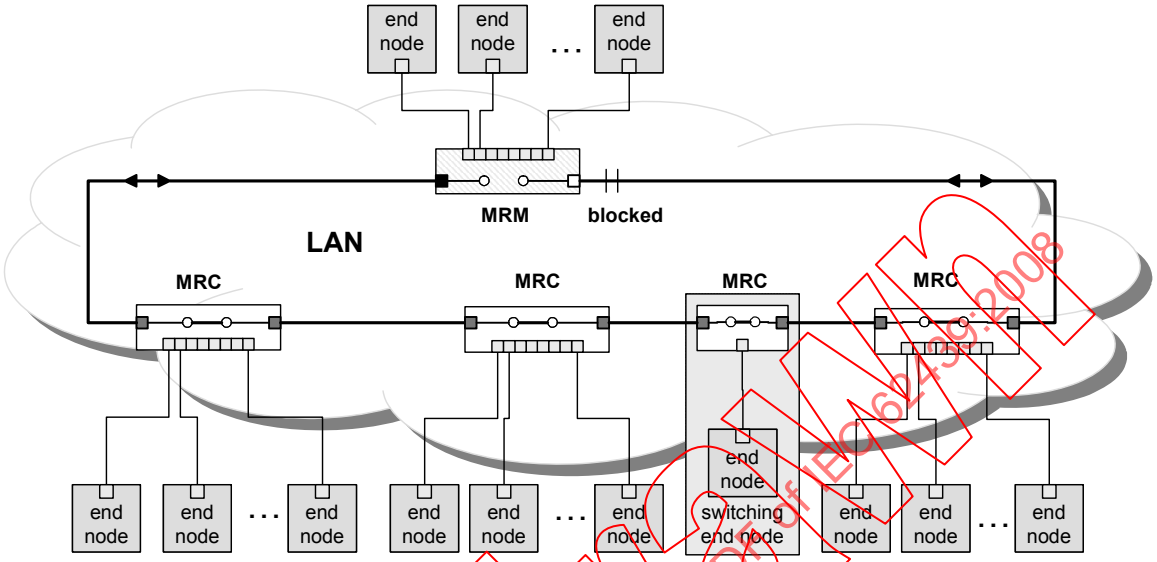


Figure 13 – MRP ring topology with one manager and clients

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 in FORWARDING state and the other ring port in BLOCKED state if it receives its own MRP_Test frames (this means that the ring is closed, see Figure 13);
- setting both ring ports in FORWARDING state if it does not receive its own MRP_Test frames within a configured time according to MRP_TSTdefaultT, MRP_TSTshortT and MRP_TSTNRmax in Table 35 (this means that the ring is open, see Figure 14).

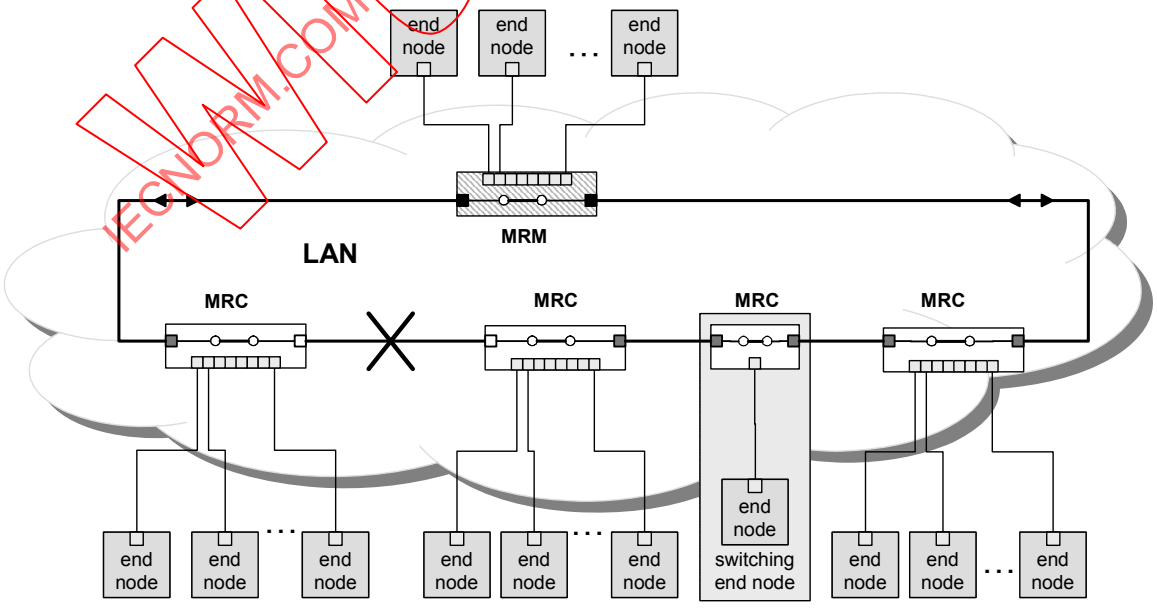


Figure 14 – MRP MRM in an open ring

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 35 to accelerate the detection of the open ring. When the open ring is detected then the MRM shall send the MRP_TopologyChange frames through both its ring ports.

Optionally, the MRM shall send the MRP_TopologyChange frames through its ring ports. This option is selected by setting the parameter REACT_ON_LINK_CHANGE; see Table 28.

The MRM shall send to the MRCs an MRP_TopologyChange frame with 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).

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 to DISABLED (MRP_LinkDown frame).

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

5.2.3 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 (see Table 35, MRP_TOPchgT).

5.2.4 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 as a key attribute, especially if an MRM or an MRC is member of multiple rings. A node shall assign exactly two unique ring ports per redundancy domain.

NOTE 1 A device may have other ports than the two assigned to MRP. These other ports are not influenced by MRP.

NOTE 2 MRP ports should behave as if RSTP is disabled.

5.2.5 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.

5.2.6 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 signalled by using the State Change service see 5.4.3.

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

5.2.7 Multiple MRM in a single ring

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

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

As an option, in case of more than one node having the ability to become an MRM in the ring, an enhanced protocol not specified in this standard may be used to decide which of these nodes shall become the MRM, while the other nodes take over the MRC role as shown in Figure 15. 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 optional protocol for multiple MRM in a single ring is used then all MRM in the ring shall support the same protocol. The vendor shall specify the supported protocols.

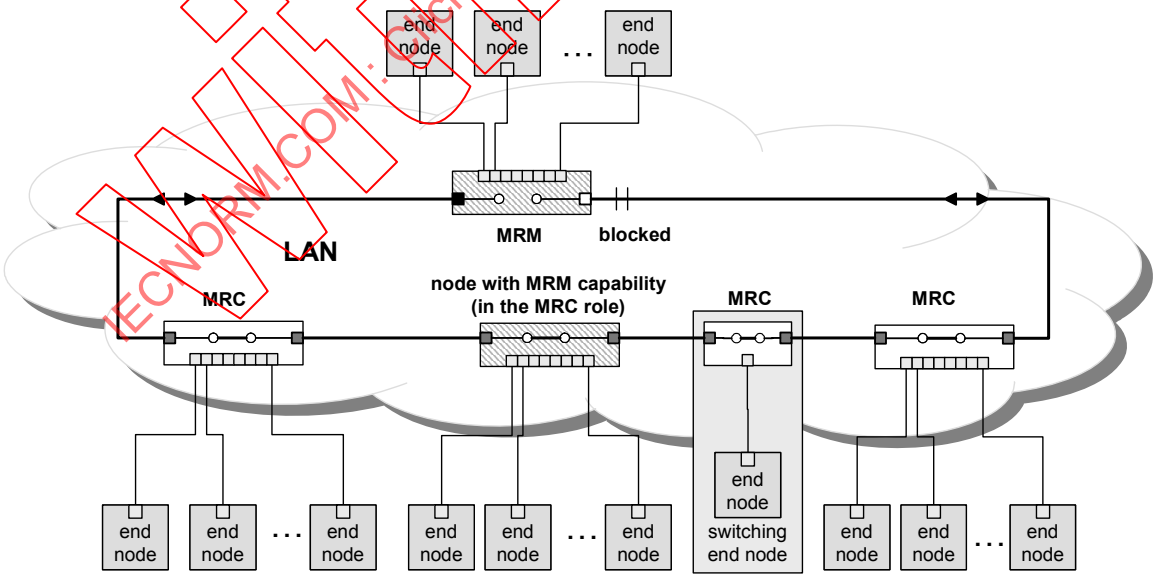


Figure 15 – MRP ring with more than one MRM

5.2.8 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 a MRM shall support additional functionalities (see Table 28, MRP_BLOCKED_SUPPORTED).

5.3 MRP Class specification

5.3.1 General

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

5.3.2 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, 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

5.3.3 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, CLIENT

Manager Priority
This attribute shall contain the priority of the MRM. A lower value indicates a higher priority, 0x0000 (highest priority) to 0xF000 (lowest priority) in increments of 0x1000.

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 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, 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 5.2.1.

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 5.2.1.

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

5.4 MRP Service specification

5.4.1 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

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

5.4.2 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**

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

5.4.3 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**

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

5.4.4 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

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

5.4.5 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

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

5.4.6 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
CheckMediaRedundancy		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

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

5.4.7 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

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

5.5 MRP Protocol specification

5.5.1 PDU description

5.5.1.1 Basic data types

The conventions for this specification are according IEC 61158-6-10, 3.6. Notation and encoding of basic data types according to IEC 61158-6-10, 4.1.2 and IEC 61158-6-10, 4.2.

5.5.1.2 DLPDU abstract syntax reference

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

The encoding and decoding of the fields in Table 10 shall be according to IEEE 802.3 for the DLPDU.

Table 10 – MRP IEEE 802.3 DLPDU syntax

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 IEEE 802.3 the DLPDUs have a minimum length of 64 octets (excluding Preamble and Start Frame Delimiter). NOTE 2 For IEEE 802.3 frames with VLAN tag the minimum frame size is increased to 68 octets in order to guaranty 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.	

5.5.1.3 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.

5.5.1.4 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 11.

Table 11 – 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 12.

Table 12 – 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 and MRP_TopologyChange frames
01-15-4E	00-00-03 to FF-FF-FF	Reserved

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

5.5.1.5 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.CanonicalFormatIdentificator
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 13.

Table 13 – MRP TagControlInformation.Priority field

Value (hexadecimal)	Meaning
0x07	MRP-PDU

5.5.1.6 Coding of the field LT

This field shall be coded as data type Unsigned16 with the values according to IEEE 802.3. For MRP the value shall be set according to Table 14.

Table 14 – MRP LT field

Value (hexadecimal)	Meaning
0x88E3	MRP-PDU

5.5.1.7 MRP APDU abstract syntax

Table 15 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 15 – 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 IEEE 802.3.	

Table 16 defines structures for substitutions of elements of the APDU structure shown in Table 15.

Table 16 – MRP Substitutions

Substitution name	Structure
MRP_Type	MRP_Test ^ MRP_LinkChange ^ MRP_TopologyChange ^ MRP_Option
MRP_Common	MRP_TLVHeader, MRP_SequenceID, MRP_DomainUUID
MRP_Option	MRP_TLVHeader, MRP_ManufacturerOUI, MRP_ManufacturerData, [Padding*][a]
MRP_End	MRP_TLVHeader (=0x0000)
MRP_Test	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]
[a] 32-bit alignment shall be ensured.	

5.5.1.8 Coding of the field MRP_TLVHeader

The coding of this field shall be according to IEC 61158-6-10, 3.6.3.4 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 17.

Table 17 – 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 – 0x7E	Reserved	—
0x7F	MRP_Option (Organizationally specific)	Optional
0x80-0xFF	Reserved	—

5.5.1.9 Coding of the field MRP_Version

This field shall be coded as data type Unsigned16 with the values according to Table 18.

Table 18 – MRP_Version

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

5.5.1.10 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.

5.5.1.11 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).

5.5.1.12 Coding of the field MRP_Prio

This field shall be coded as data type Unsigned16 and set according to Table 19.

Table 19 – MRP_Prio

Value (hexadecimal)	Meaning
0x0000	Highest priority redundancy manager
0x1000 – 0x7000	High priorities
0x8000	Default priority for redundancy manager
0x9000 – 0xE000	Low priorities
0xF000	Lowest priority redundancy manager
Other	Reserved

5.5.1.13 Coding of the field MRP_PortRole

This field shall be coded as data type Unsigned16. The coding shall be according to Table 20.

Table 20 – 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 – 0xFFFF	Reserved	

5.5.1.14 Coding of the field MRP_RingState

This field shall be coded as data type Unsigned16 with the values according to Table 21.

Table 21 – 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	—

5.5.1.15 Coding of the field MRP_Interval

This field shall be coded as data type Unsigned16 with the values according to Table 22.

Table 22 – 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

5.5.1.16 Coding of the field MRP_Transition

This field shall be coded as data type Unsigned16 with the values according to Table 23.

Table 23 – MRP_Transition

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

5.5.1.17 Coding of the field MRP_TimeStamp

This field shall be coded as data type Unsigned32 with the values according to Table 24.

Table 24 – 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.

5.5.1.18 Coding of the field MRP_Blocked

This field shall be coded as data type Unsigned16 with the values according to Table 25.

Table 25 – 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	—

5.5.1.19 Coding of the field MRP_ManufacturerOUI

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

5.5.1.20 Coding of the field MRP_ManufacturerData

This field shall be reserved for vendor-specific data.

5.5.1.21 Coding of the field MRP_DomainUUID

This field shall be coded as UUID with the values according to Table 26.

Table 26 – MRP_DomainUUID

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

5.5.2 Protocol machines

5.5.2.1 MRM Protocol machine

The MRM protocol machine is defined in Table 28. The principal behaviour of the protocol machine is shown in Figure 16.

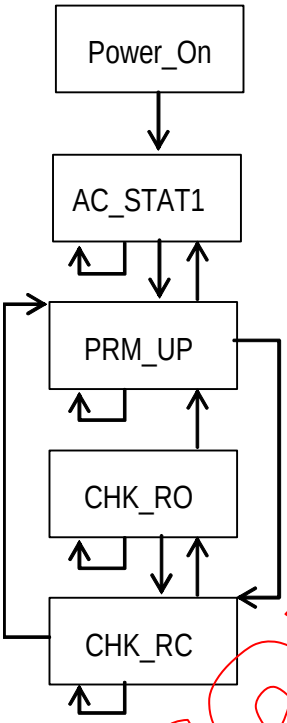


Figure 16 – MRP Protocol machine for MRM

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 and MC_CONTROL to host are generated. All MRP-PDU shall use the highest priority (ORG).

AC_STAT1
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 set to ring closed state.

Local variables of the MRM protocol machine are listed in Table 27.

Table 27 – 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
AddTest	Boolean	Send additional MRP-PDU of type MRP_Test after MRP_TSTshortT intervall if TRUE
ReactMode	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
NoTC	Boolean	Suppress MRP_TopologyChange while in line topology (NoTC = TRUE)

The MRM state machine shall be according to Table 28.

Table 28 – MRM state machine

#	Current state	Event /condition =>action	Next state
1	Power On	=> INIT_FDB ADD_MAC_FDB({local},{MC_TEST, MC_CONTROL},ORG) PRM_RPort:= RPort_1 SEC_RPort:= RPort_2 MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 AddTest:= FALSE ReactMode:= 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	**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** => AddTest:= 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 NoTC:= 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 NoTC:= 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)** /!AddTest && PortMode == MRP_BLOCKED_SUPPORTED => AddTest:= TRUE TestRingReq(MRP_TSTshortT)	PRM_UP
16	PRM_UP	**LinkChangeInd(PortMode, Link_status)** /AddTest && 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)** /AddTest && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => TopologyChangeReq(0)	PRM_UP
19	PRM_UP	**LinkChangeInd(PortMode, Link_status)** /!AddTest && Link_status == MRP_LNK_UP && PortMode != MRP_BLOCKED_SUPPORTED => AddTest:= TRUE TestRingReq(MRP_TSTshortT) TopologyChangeReq(0)	PRM_UP
20	PRM_UP	**TopologyChangeInd(MRP_SA, t)** => ignore	PRM_UP
21	CHK_RO	**TestTimer expired** => AddTest:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RO
22	CHK_RO	**MAUType_ChangeInd (RPort, Link_status)** /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RO

#	Current state	Event /condition =>action	Next state
23	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
24	CHK_RO	**MAUType_ChangeInd (RPort, Link_status)** /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RO
25	CHK_RO	**MAUType_ChangeInd (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 && ReactMode != MRP_REACT_ON_LINK_CHANGE => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NoTC:= FALSE TestRingReq(MRP_TSTdefaultT) TopologyChangeReq(MRP_TOPchgT)	CHK_RC
27	CHK_RO	**TestRingInd(MRP_SA, MRP_Prio)** /MRP_SA == MRP_TS_SA && ReactMode == MRP_REACT_ON_LINK_CHANGE => Set_Port_StateReq (SEC_RPort, BLOCKED) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 NoTC:= 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)** /!AddTest && Link_status == MRP_LNK_UP && PortMode == MRP_BLOCKED_SUPPORTED => AddTest:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RO
30	CHK_RO	**LinkChangeInd(PortMode, Link_status)** /AddTest && Link_status == MRP_LNK_UP && PortMode == MRP_BLOCKED_SUPPORTED => Ignore	CHK_RO
31	CHK_RO	**LinkChangeInd(PortMode, Link_status)** /AddTest && Link_status == MRP_LNK_DOWN => Ignore	CHK_RO
32	CHK_RO	**LinkChangeInd(PortMode, Link_status)** /!AddTest && Link_status == MRP_LNK_DOWN => AddTest:= TRUE TestRingReq(MRP_TSTshortT)	CHK_RO
33	CHK_RO	**LinkChangeInd(PortMode, Link_status)** /AddTest && 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)** /!AddTest && 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 AddTest:= 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 && !NoTC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 AddTest:= FALSE TopologyChangeReq(MRP_TOPchgT) TestRingReq(MRP_TSTdefaultT)	CHK_RO
37	CHK_RC	TestTimer expired /MRP_MRM_NReturn >= MRP_MRM_Nrmax && NoTC => Set_Port_StateReq (SEC_RPort, FORWARDING) MRP_MRM_NRmax:= MRP_TSTNRmax – 1 MRP_MRM_NReturn:= 0 AddTest:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RO
38	CHK_RC	**TestTimer expired** /MRP_MRM_NReturn < MRP_MRM_NRmax => MRP_MRM_NReturn:= MRP_MRM_NReturn + 1 AddTest:= FALSE TestRingReq(MRP_TSTdefaultT)	CHK_RC
39	CHK_RC	**MAUType_ChangeInd (RPort, Link_status)** /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RC

#	Current state	Event /condition =>action	Next state
40	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
41	CHK_RC	**MAUType_ChangeInd (RPort, Link_status)** /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	CHK_RC
42	CHK_RC	**MAUType_ChangeInd (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 NoTC:= 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)** /AddTest && ReactMode != MRP_REACT_ON_LINK_CHANGE && PortMode == MRP_BLOCKED_SUPPORTED => ignore	CHK_RC
46	CHK_RC	**LinkChangeInd(PortMode, Link_status)** /!AddTest && ReactMode != MRP_REACT_ON_LINK_CHANGE && PortMode == MRP_BLOCKED_SUPPORTED => AddTest:= 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 && ReactMode == 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 && ReactMode == 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 && ReactMode == 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

5.5.2.2 MRC Protocol Machine

The MRC protocol machine is defined in Table 30. The principal behaviour of the protocol machine is shown in Figure 17.

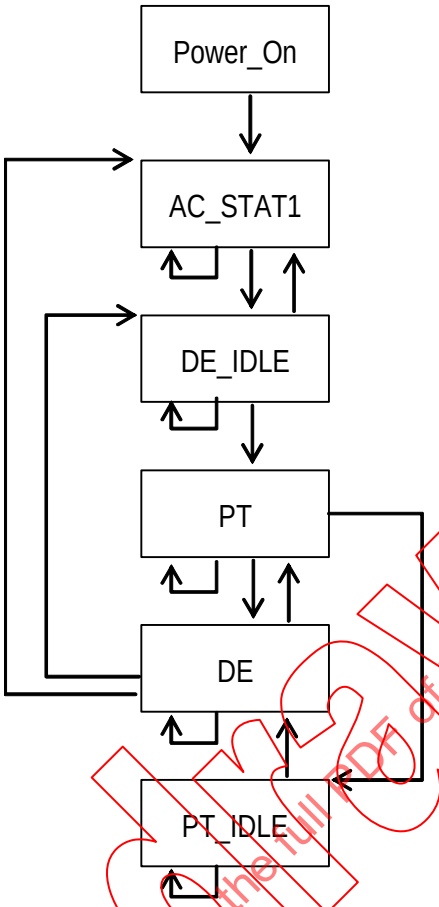


Figure 17 – MRP protocol machine for MRC

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
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 signalling link changes.

DE (Data Exchange)
Temporary state while signalling 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 29.

Table 29 – 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 30.

Table 30 – MRC State machine

#	Current state	Event /condition =>action	Next state
1	Power On	=> INIT_FDB ADD_MAC_FDB({RPort_1,RPort_2}, {MC_TEST, MC_CONTROL},ORG) ADD_MAC_FDB({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	**MAUType_ChangeInd (RPort, Link_status)** /RPort == PRM_RPort && Link_status == MRP_LNK_UP => Set_Port_StateReq (PRM_RPort, FORWARDING)	DE_IDLE
3	AC_STAT1	**MAUType_ChangeInd (RPort, Link_status)** /Link_status == MRP_LNK_DOWN => ignore	AC_STAT1

#	Current state	Event /condition =>action	Next state
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)	DE_IDLE
5	AC_STAT1	**TopologyChangeInd (MRP_SA, t)** => ignore	AC_STAT1
6	DE_IDLE	**MAUType_ChangeInd (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
7	DE_IDLE	**MAUType_ChangeInd (RPort, Link_status)** /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE_IDLE
8	DE_IDLE	**MAUType_ChangeInd (RPort, Link_status)** /RPort == PRM_RPort && Link_status == MRP_LNK_DOWN => Set_Port_StateReq (PRM_RPort, BLOCKED)	AC_STAT1
9	DE_IDLE	**MAUType_ChangeInd (RPort, Link_status)** /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	DE_IDLE
10	DE_IDLE	**TopologyChangeInd(MRP_SA, t)** => CLEAR_FDB(t)	DE_IDLE

#	Current state	Event /condition =>action	Next state
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	**MAUType_ChangeInd (RPort, Link_status)** /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT
14	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) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn X MRP_LNKdownT)	DE
15	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 (SEC_RPort, BLOCKED) DownTimer.start(MRP_LNKdownT) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn X MRP_LNKdownT)	DE

#	Current state	Event /condition =>action	Next state
16	PT	**MAUType_ChangeInd (RPort, Link_status)** /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT
17	PT	**TopologyChangeInd(MRP_SA, t)** => MRP_LNKNReturn:= MRP_LNKNRmax UpTimer.stop Set_Port_StateReq (SEC_RPort, FORWARDING) CLEAR_FDB(t)	PT_IDLE
18	DE	**DownTimer expired** /MRP_LNKNReturn == 0 => MRP_LNKNReturn:= MRP_LNKNRmax	DE_IDLE
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	**MAUType_ChangeInd (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	**MAUType_ChangeInd (RPort, Link_status)** /RPort != PRM_RPort && Link_status == MRP_LNK_DOWN => ignore	DE

#	Current state	Event /condition =>action	Next state
22	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	AC_STAT1
23	DE	**MAUType_ChangeInd (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 CLEAR_FDB(t)	DE_IDLE
25	PT_IDLE	**MAUType_ChangeInd (RPort, Link_status)** /RPort != PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
26	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) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn X MRP_LNKdownT)	DE

#	Current state	Event /condition =>action	Next state
27	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) LinkChangeReq(PRM_RPort, MRP_LNK_DOWN, MRP_LNKNReturn X MRP_LNKdownT)	DE
28	PT_IDLE	**MAUType_ChangeInd (RPort, Link_status)** /RPort == PRM_RPort && Link_status == MRP_LNK_UP => ignore	PT_IDLE
29	PT_IDLE	**TopologyChangeInd(MRP_SA, t)** => CLEAR_FDB(t)	PT_IDLE

5.5.2.3 MRM and MRC Functions

The MRM and MRC functions shall be according to Table 31.

Table 31 – MRP functions

Function name	Operations
TestRingReq(t)	SETUP_TEST_RING_REQ TestTimer.start(t)

Function name	Operations
SETUP_TEST_RING_REQ	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 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)	SETUP_TOPOLOGY_CHANGE_REQ (MRP_TOPNRmax X time) if time == 0 CLEAR_LOCAL_FDB else TopTimer.start(MRP_TOPchgT)
SETUP_TOPOLOGY_CHANGE_REQ (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

Function name	Operations
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_TEST, 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
MAUType_ChangeInd(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
CLEAR_FDB(time)	FDBClearTimer.start(time)
CLEAR_LOCAL_FDB	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
INIT_FDB	Function to initialize Filtering Data Base
ADD_MAC_FDB(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

5.5.2.4 FDB Clear Timer

FDB Clear Timer is an auxiliary state machine. FDB Clear Timer shall be according to the state machine in Table 32.

Table 32 – MRP FDB clear timer

#	Current state	Event /condition =>action	Next state
1	Power On	=> FDBClearTimer.ini	IDLE
2	IDLE	**FDBClearTimer.expired** => CLEAR_LOCAL_FDB	IDLE

5.5.2.5 Topology Change Timer

Topology Change Timer is an auxiliary state machine. Topology Change Timer shall be according to the state machine in Table 33.

Table 33 – MRP topology change timer

#	Current state	Event /condition =>action	Next state
1	Power On	=> TopTimer.ini TC_NReturn:= MRP_TOPNRmax - 1	IDLE
2	IDLE	**TopTimer expired** /TC_NReturn > 0 => SETUP_TOPOLOGY_CHANGE_REQ (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) SETUP_TOPOLOGY_CHANGE_REQ (0)	IDLE

5.6 MRP Installation, configuration and repair

5.6.1 Ring port parameters

Ring port parameterization for the MRM and all MRC in a ring shall comply with the settings from Table 34.

Table 34 – MRP network/connection parameters

Parameter	Value
Link speed	The link speed shall be at least 100 Mbit/s
Duplex setting	Ring ports 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

5.6.2 Ring topology parameters

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

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

5.6.3 MRM and MRC parameters

The MRM defines with its parameter set the recovery time of a ring. Table 35 and Table 36 specify two consistent sets of parameters for a ring recovery time of 500 ms and a ring recovery time of 200 ms.

NOTE Additional consistent parameter sets for shorter or longer maximum recovery times may be supported in an MRM and MRC. The designer is responsible for the consistency for all parameter in 5.6, the installer is responsible for the consistency of all nodes in the ring.

Table 35 – MRP MRM parameters

Parameter	Max. recovery time		Meaning
	500 ms	200 ms	
MRP_TOPchgT	20 ms	10 ms	Topology Change (Clear Address Table) request interval
MRP_TOPNRmax	3	3	Topology Change (Clear Address Table) repeat count
MRP_TSTshortT	30 ms	10 ms	MRP_Test short interval
MRP_TSTdefaultT	50 ms	20 ms	MRP_Test default interval
MRP_TSTNRmax	5	3	MRP_Test monitoring count
MRP_TSTExtNRmax	15	Not applicable[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 = 200 ms. All MRCs shall support blocking mode. MRP_TSTExtNRmax not applicable.

Table 36 specifies the MRC parameter sets (used for both MRM parameter sets).

Table 36 – MRP MRC parameters

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

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

5.6.4 Configuration

For the configuration of MRP nodes there are two optional network management information bases (MIBs) provided in Clause C.2.

6 PRP – Parallel Redundancy Protocol

6.1 PRP Principle of operation

This redundancy protocol implements redundancy in the devices. An end node is attached to two independent LANs of similar topology, which are operated in parallel.

The two LANs follow configuration rules that allow the network management protocols such as address resolution protocol (ARP) to operate correctly.

Figure 18 shows a redundant network consisting of two switched LANs, which can have any topology, for example, tree, ring or meshed.

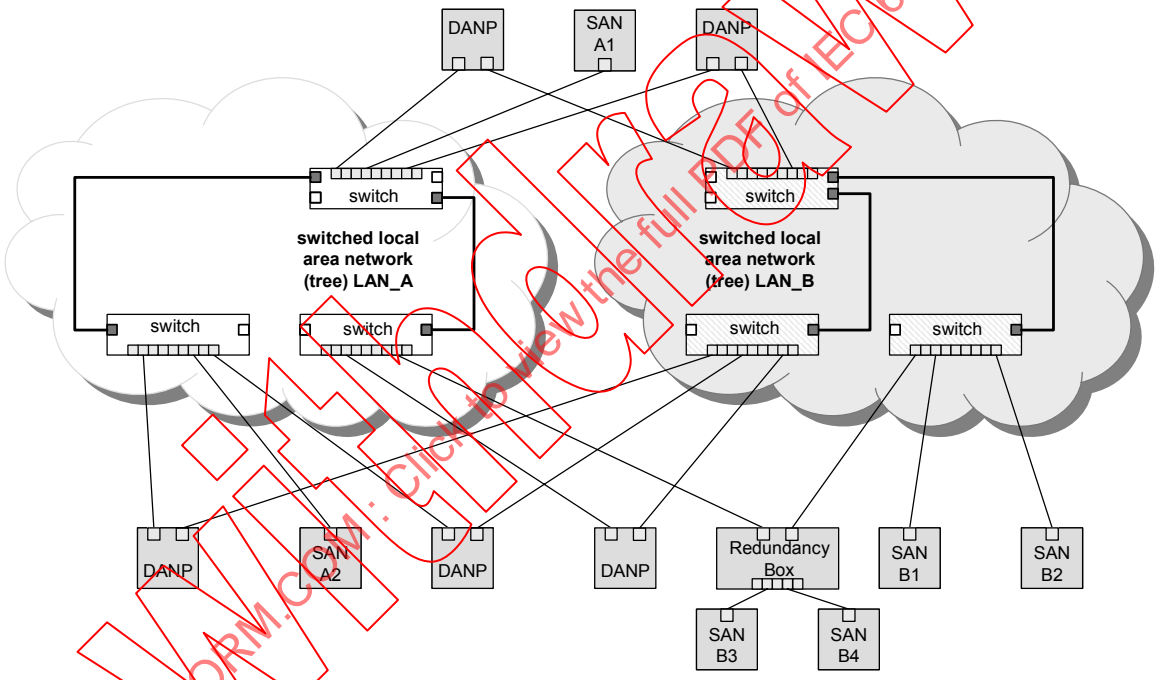


Figure 18 – PRP general redundant network example

The two LANs are identical in protocol at the MAC-LLC level, but they can differ in performance and topology. Transmission delays may also be different. The LANs have no direct connection between them and they are assumed to be fail-independent.

A doubly attached node implementing PRP (DANP) is attached to both LANs, named LAN_A and LAN_B.

Singly attached nodes (SANs) can be attached in two ways:

- SANs can be attached direct to one LAN only. SANs can only communicate with other SANs on the same LAN. For instance, in Figure 18, SAN A1 can communicate with SAN A2, but not with SAN B1 or SAN B2. SANs can communicate with all DANPs.
- SANs can be attached over a redundancy box to both LANs, as Figure 18 shows (see also 6.1.5). Such SANs can communicate with all SANs, for instance SAN A1 and SAN B3 can communicate.

NOTE SANs do not need to be aware of PRP, they can be off-the-shelf computers.

As an example of a simpler configuration, Figure 19 draws a PRP network as two LANs in linear topology, which may also be a bus topology.

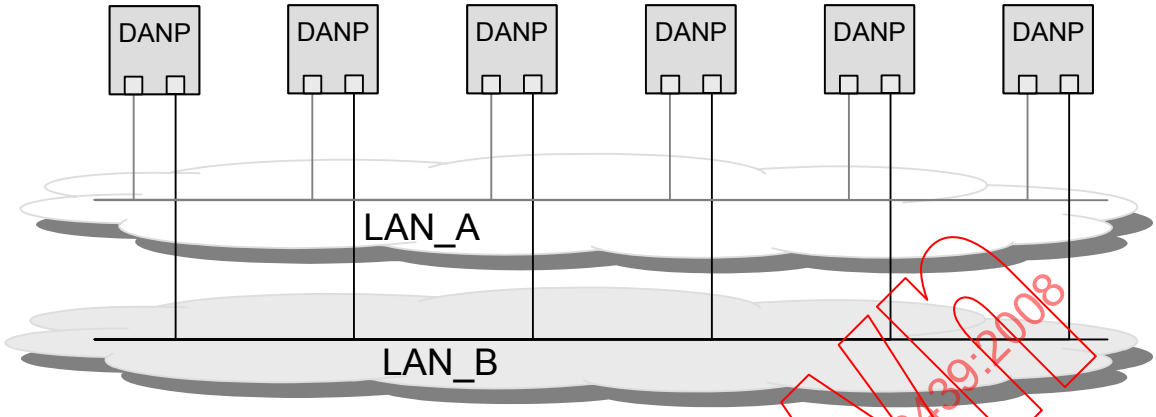


Figure 19 – PRP Redundant network example as two LANs (bus topology)

The two LANs can have a ring topology, as Figure 20 shows.

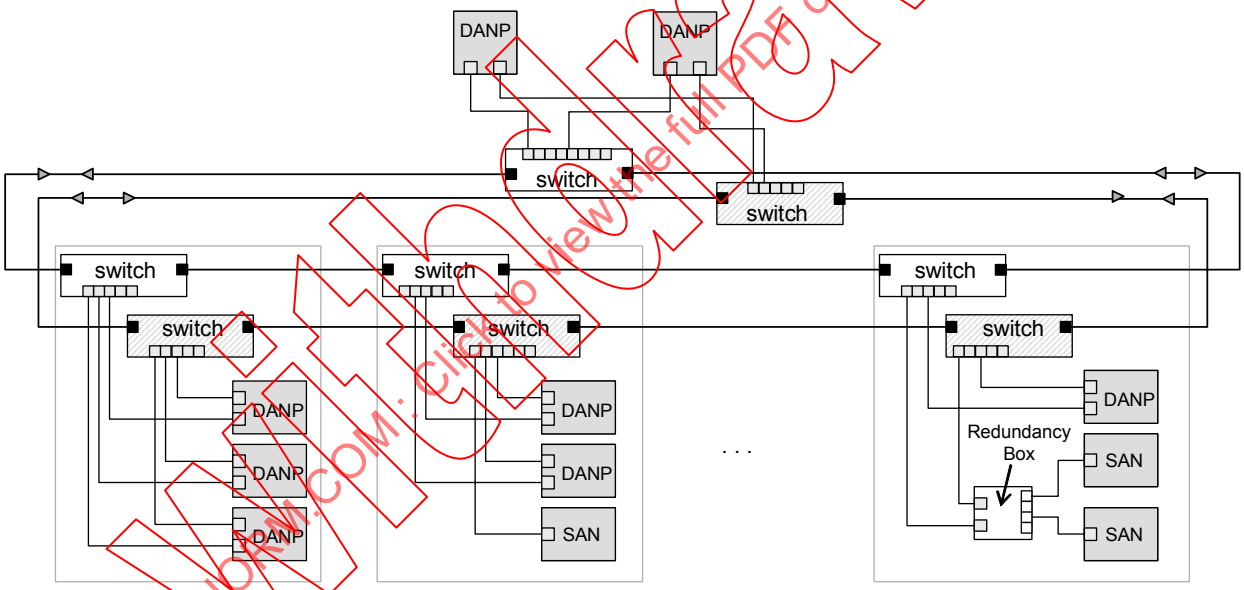


Figure 20 – PRP redundant ring example with SANs and DANPs.

In some applications, only availability-critical devices need a double attachment, for instance the operator workplaces, while the majority of the devices are SANs, as Figure 21 shows. Taking advantage of the basic infrastructure of PRP, a DANP can be attached to two different switches of the same LAN (for example, a ring) and use protocols different from PRP to reconfigure the network in case of failure. The DANP then behaves as a switch element according to IEEE 802.1D. For instance, the switch element may implement the MRP protocol, the RSTP protocol, or a subset of RSTP, called Serial Redundancy Protocol (SRP) in which the node behaves like a RSTP switch element, except that it does not forward traffic between its ports. These abilities are optional and not detailed in this standard. The supported mode is specified in the PICS (see 6.4).

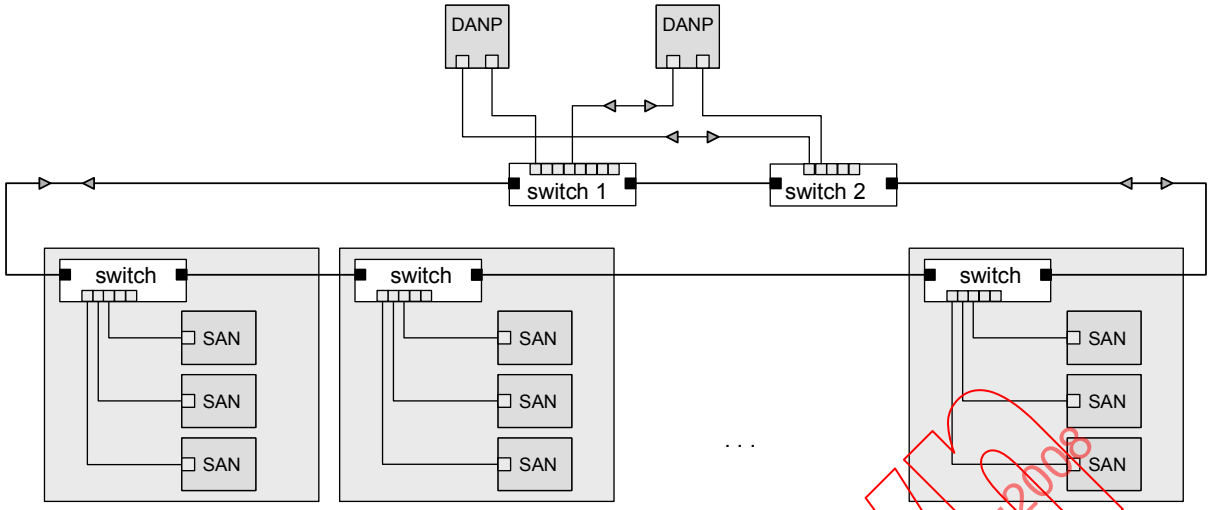


Figure 21 – PRP single ring with DANPs in SRP mode

6.1.1 Single points of failure

The LANs are assumed to be fail-independent. Redundancy can be defeated by single points of failure, such as a common power supply or a direct connection whose failure brings both networks down. Installation guidelines in this standard provide guidance to the installer to achieve fail-independence.

6.1.2 Node structure

Each node has two ports that operate in parallel and that are attached to the same upper layers of the communication stack through the LRE, as Figure 22 shows.

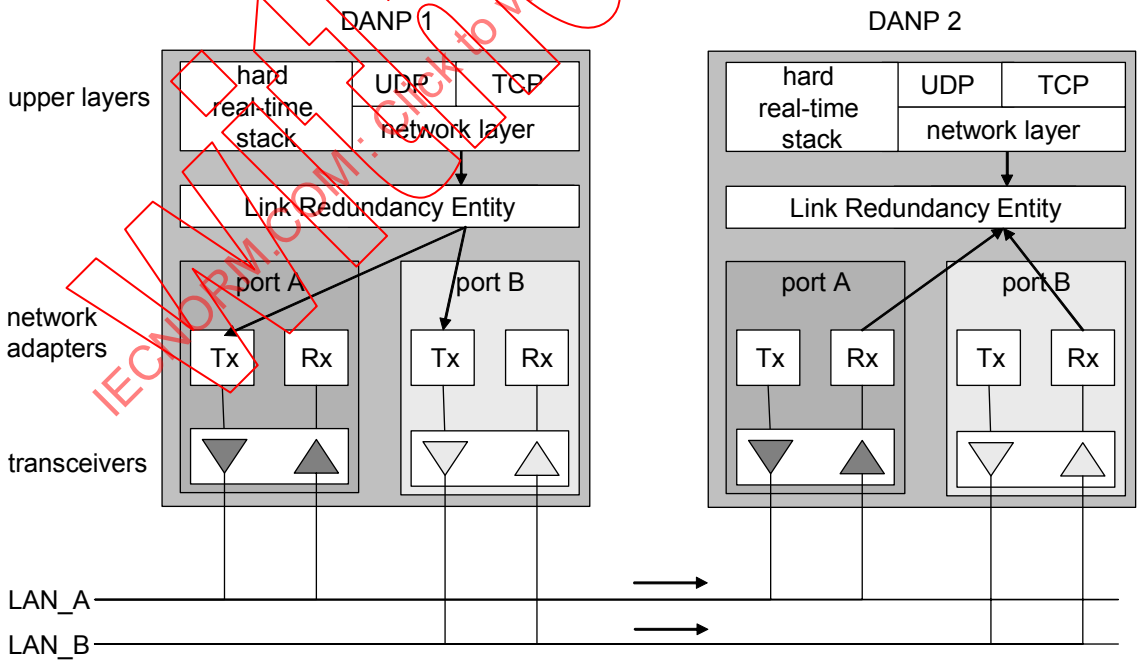


Figure 22 – PRP two DANPs communicating

The LRE has two tasks: handling of duplicates and management of redundancy. This layer presents toward its upper layers the same interface as the network adapter of a non-redundant adapter.

When receiving a frame from the node's upper layers, the LRE sends the frame through both its ports at nearly the same time.

The two frames transit through the two LANs with different delays, ideally they arrive at the same time at the destination node.

When receiving frames from the network, the LRE forwards the first received frame of a pair to the node's upper layers and discards the duplicate frame (if it arrives).

For management of redundancy, the LRE can append a redundancy check trailer (RCT) including a sequence number to the frames it sends to keep track of duplicates. In addition, the LRE periodically sends PRP_Supervision frames and evaluates the PRP_Supervision frames of the other DANPs.

6.1.3 Compatibility between singly and doubly attached nodes

Singly attached nodes (SAN), for instance maintenance laptops or printers that belong to one LAN, can be connected to any LAN. A SAN connected to one LAN cannot communicate directly to a SAN connected to the other LAN. Switches are always SANs. These SANs are not aware of PRP redundancy, so DANPs generate a traffic that these SANs understand. The condition is however that the SANs ignore the RCT in the frames, which should be the case since a SAN cannot distinguish the RCT from IEEE 802.3 padding. Conversely, DANPs understand the traffic generated by SANs, since these do not append an RCT. They only forward one frame to their upper layers since the SAN traffic uses one LAN only. If a DANP cannot positively identify that the remote device is a DANP, it considers it as a SAN.

6.1.4 Network management

A node has the same MAC address on both ports, and only one set of IP addresses assigned to that address. This makes redundancy transparent to the upper layers. Especially, this allows the address resolution protocol (ARP) to work in the same way as with a SAN. Switches in a LAN are not doubly attached devices, and, therefore, all managed switches have different IP addresses. A network management tool is preferably a DANP and can access nodes and switches as if they all belong to the same network. Especially, network management implemented in a DANP is able to see SANs connected to either LAN.

Some applications require different MAC addresses on the redundant ports, and these MAC addresses may be different from the default MAC address of that node. This involves address substitution mechanisms which are not specified in this standard. However, the basic protocol and the frame format are prepared for such extension. Nodes that support MAC address substitution are indicated as supporting PICS_SUBS.

6.1.5 Transition to non-redundant networks

The mechanism of duplicate rejection can be implemented in a device called a redundancy box that does the transition between a SAN and the doubled LANs, as Figure 23 shows. The redundancy box mimics the SANs connected behind it (called VDA or virtual DANs) and multicasts supervision frames on their behalf, appending its own information. The redundancy box is itself a DANP and has its own IP address for management purposes but it may also perform application functions.

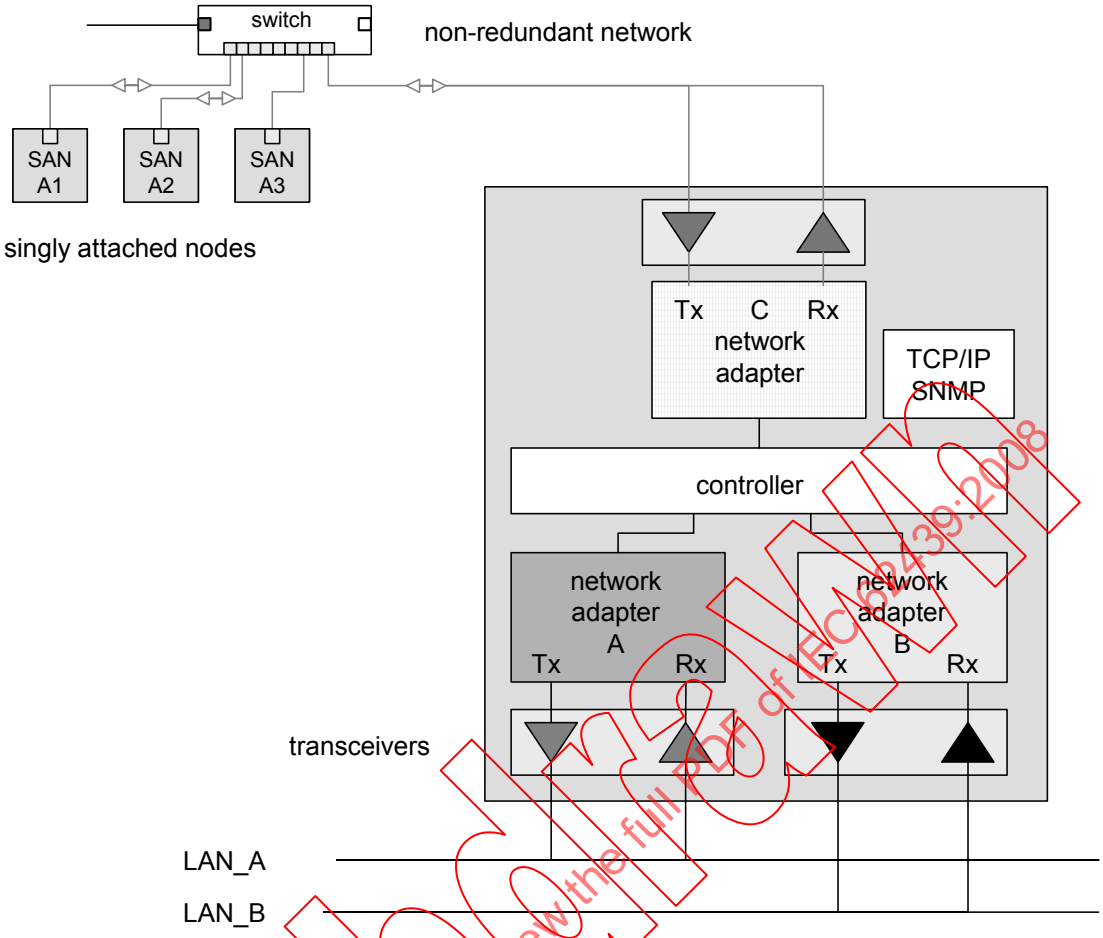


Figure 23 – PRP redundancy box, transition from single to double LAN.

6.1.6 Duplicate handling

6.1.6.1 Methods for handling duplicates

Since a DANP receives the same frame over both adapters, when both are operational, it should keep one and ignore the duplicate.

There are two methods for handling duplicates.

a) Duplicate accept, in which the sender LRE uses the original frames and the receiver LRE forwards both frames it receives to its upper protocol layers.

b) Duplicate discard, in which the sender LRE appends a redundancy control trailer to both frames it sends and the receiver LRE uses that redundancy control trailer to send only the first frame of a pair to its upper layers and filter out duplicates.

6.1.6.2 Duplicate accept

This method does not attempt to discard duplicates at the link layer. The sender LRE sends the same frame as it would in the non-redundant case over both LANs. The receiver's LRE forwards both frames of a pair (if both arrive) to its upper layers, assuming that well-designed network protocols and applications are able to withstand duplicates – indeed IEEE 802.1D explicitly states that it cannot ensure freedom of duplicates.

The internet stack, consisting of a network layer with an UDP and a TCP transport layer, is assumed to be resilient against duplicates. The TCP protocol is designed to reject duplicates, so it discards the second frame of a pair. The UDP layer is by definition connectionless and unacknowledged. All applications that use UDP are assumed to be capable of handling duplicates, since duplication of frames can occur in any network. In particular, a UDP frame is

assumed to be idempotent, i.e. sending it twice has the same effect as sending it once. Administrative protocols of the internet such as ICMP and ARP are not affected by duplicates, since they have their own sequence numbering.

Real-time stack that operate on the publisher-subscriber principle are not affected by duplicates, since only the latest value is kept. Duplicate reception increases robustness since a sample that gets lost on one LAN is usually received from the other LAN.

Therefore, one can assume that handling of duplicates is taken care of by the usual network protocols, but one has to check if each application complies with these assumptions.

This simple duplicate accept method does not provide easy redundancy supervision, since it does not keep track of correct reception of both frames. The receiver would need hash tables to know that a frame is the first of a pair of a duplicate, and could for this effect store the CRC and length of each frame as a hash code. Such redundancy supervision method is, however, not specified in this standard, but it is not excluded.

6.1.6.3 Duplicate discard in the link layer

6.1.6.3.1 Principle

It is advantageous to discard duplicates already at the link layer.

Without duplicate discard, the processor receives twice as many interrupt requests as when only one LAN is connected. To offload the application processor, the LRE can perform duplicate discard, possibly with an independent pre-processor or an intelligent Ethernet controller. This allows at the same time to improve the redundancy supervision.

The duplicate discard protocol uses an additional four-octet field in the frame, the RCT, which the LRE inserts into each frame that it receives from the upper layers before sending, as Figure 24 shows. The RCT consists of the following parameters:

a) 16-bit sequence number (SequenceNr)

b) 4-bit LAN identifier (Lan)

c) 12-bit frame size (LSDU_size)

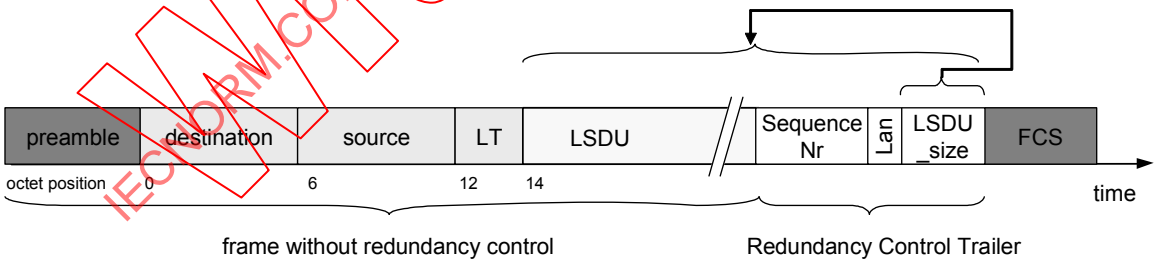


Figure 24 – PRP frame extended by an RCT

6.1.6.3.2 Use of SequenceNr

Each time an LRE sends a frame to a particular destination, it increases the sequence number corresponding to that destination and sends both (nearly identical) frames over both LANs.

The receiving LRE can then detect duplicates based on the RCT.

This method considers that SANs also exist on the network, and that frames sent by SANs could be wrongly rejected as duplicates because they happen to have a trailing field with the same sequence number and the same size. However, SANs send on one LAN only, and the

source will not be the same as that of another frame, so a frame from an SAN will never be discarded.

6.1.6.3.3 Use of Lan

The field Lan can take one of two values: 1010 indicating that the frame has been sent over LAN_A and 1011 indicating that the frame has been sent over LAN_B. This allows detecting installation errors.

6.1.6.3.4 Use of LSDU_size

To allow the receiver LRE to distinguish easily frames coming from nodes that obey to the PRP from the non-redundant ones, the sender LRE appends to the frame the length of the link service data unit (LSDU) in octets in a 12-bit field.

EXAMPLE If the frame carries a 100-octets LSDU, the size field equals LSDU+RCT: 104 = 100 + 4.

In VLANs, frame tags may be added or removed during transit through a switch. To make the length field independent of tagging, only the LSDU and the RCT are considered in the LSDU_size, as Figure 25 shows.

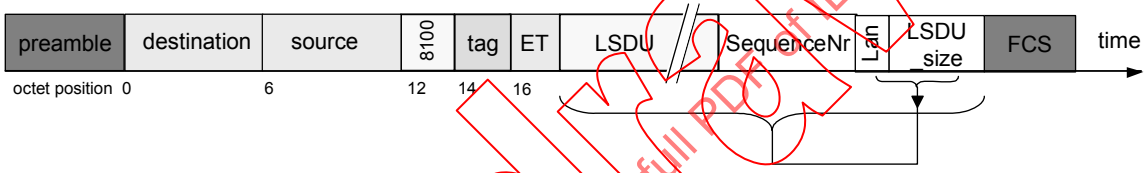


Figure 25 – PRP Tagged frame extended by an RCT

The receiver scans the frames, preferably starting from the end. If it detects that the 12 bits before the end correspond to the LSDU size, and that the LAN identifier matches the identifier of the LAN it is attached to (see 6.1.7), the frame is a candidate for rejection.

Since short frames need padding to meet the minimum frame size of 64 octets, the sender already includes the padding to speed up scanning from behind, as Figure 26 shows.

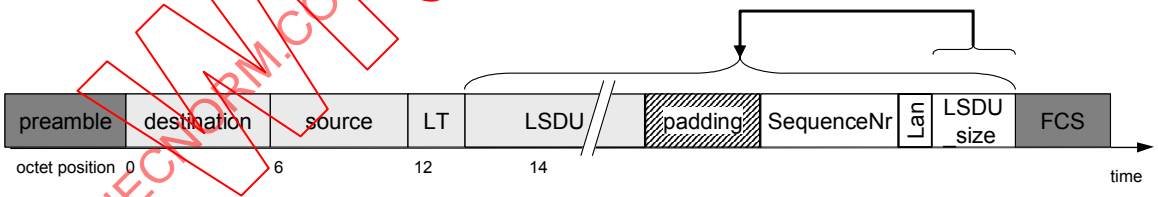


Figure 26 – PRP constructed, padded frame closed by an RCT

NOTE A tagged frame can pass several switches which may remove or insert tags. If the sender observes the IEEE 802.3 rule (repeated in Table 10) to send a minimum frame size of 68 octets for a tagged frame and of 64 for an untagged frame, there should never be a situation in which there is padding before and after the RCT. Scanning from behind is specified as a matter of precaution.

6.1.6.3.5 Frame size restriction

Appending the RCT could generate oversize frames that exceed the maxValidSize foreseen by IEEE 802.3.

To maintain compliance with IEEE 802.3-2005, the communication software in a DANP using duplicate discard is configured for a maximum payload size of 1 496 octets

NOTE Longer payloads would work in most cases, but this requires previous testing. Many switches are dimensioned for double-tagged (non-IEEE 802-3 compliant) frames that have a maximum size of 1 526 octets. Most

Ethernet controllers are certified up to 1 528 octets. Most switches would forward correctly frames of up to 1 536 octets, but this cannot be relied upon.

6.1.6.3.6 Discard algorithm

The receiver assumes that frames coming from a DANP are sent in sequence with increasing sequence numbers. The sequence number expected for the next frame is kept in the variables ExpectedSeqA, respectively ExpectedSeqB.

At reception, the correct sequence can be checked by comparing ExpectedSeqA with the received sequence number in the RCT, CurrentSeqA. Regardless of the result, ExpectedSeqA is set to one more than CurrentSeqA to allow checking the next expected sequence number on that line. The same applies to ExpectedSeqB and CurrentSeqB on LAN_B.

Both LANs thus maintain a sliding drop window of contiguous sequence numbers, the upper bound being ExpectedSeqA (the next expected sequence number on that LAN), excluding that value, the lower bound being StartSeqA (the lowest sequence number that leads to a discard on that LAN) as Figure 27 shows for LAN_A. The same applies to ExpectedSeqB and StartSeqB on LAN_B.

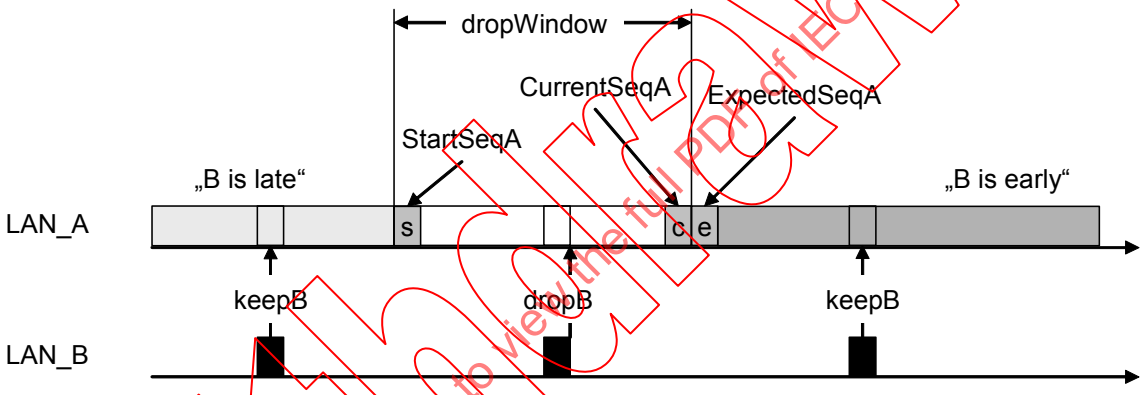


Figure 27 – PRP Drop window on LAN_A

After checking the correct sequence number, the receiver decides whether to discard the frame or not. Assuming that LAN_A has established a non-void drop window (as in Figure 27), a frame from LAN_B whose sequence number CurrentSeqB fits into the drop window of A is discarded (dropB in Figure 27). In all other cases, the frame is kept and forwarded to the upper protocol layers (keepB in Figure 27).

Discarding the frame (dropB in Figure 27) shrinks the drop window size on LAN_A since no more frames from B with an earlier sequence number are expected, thus StartSeqA is increased to one more than the received CurrentSeqB. Also, the drop window on B is reset to a size of 0 (StartSeqB = ExpectedSeqB), since obviously B lags behind A and no frames from A should be discarded, as Figure 28 shows.

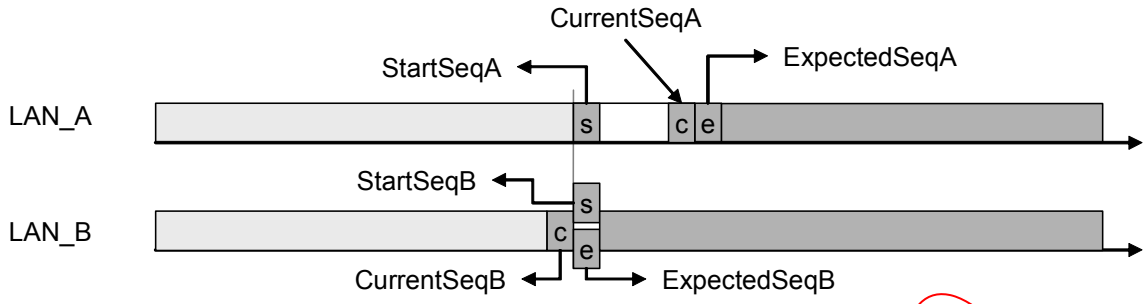


Figure 28 – PRP drop window reduction after a discard

In the situation of Figure 28, if several frames come in sequence over the same LAN_A, but none on LAN_B, they are kept since their CurrentSeqA is outside the drop window of LAN_B, and the drop window of LAN_A grows by one position. If frames keep on coming over LAN_A but not LAN_B when the maximum drop window size is reached, StartSeqA is also incremented to slide the drop window.

When a received frame is out of the drop window of the other LAN, it is kept and the drop window of that line is reduced to a size of 1, meaning that only a frame from the other line with the same sequence number is discarded, while the drop window of the other line is reset to 0, meaning that no frame is discarded, as Figure 29 shows.

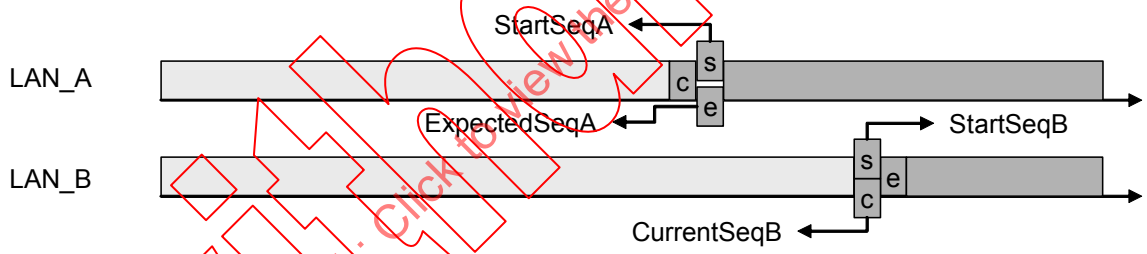


Figure 29 – PRP frame from LAN_B was not discarded.

The most common situation is when the two lines are synchronized and both drop windows are reduced to 0, meaning that the first frame to come next is kept and the drop window is opened by one to allow only a frame with the same sequence number as the one already received, as Figure 30 shows.

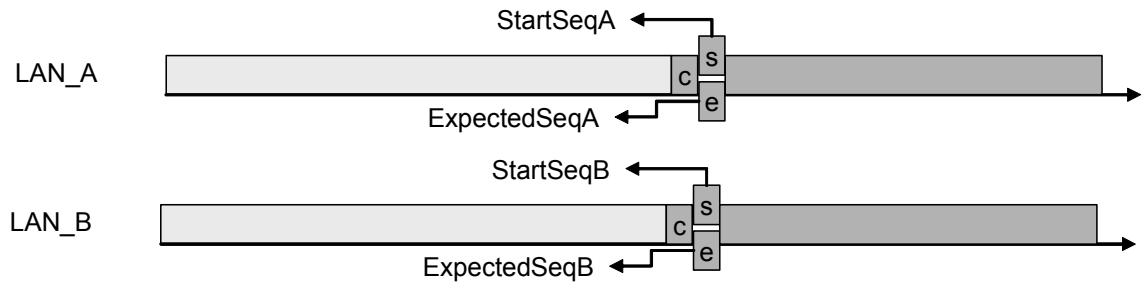


Figure 30 – PRP synchronized LANs

The sequence counter has 16 bits, which allows a drop window size of 32768, a size large enough so that even under the worst-case network delays and highest frame rate the sequence numbers do not wrap-around.

There is no change to this algorithm when frames come out of sequence.

This method can be defeated by some situations, for instance nodes failing and recovering or reconnection of a damaged LAN after a long time, but in case of doubt, duplicates are accepted so that no frame is lost.

Annex D discloses a pseudo-code for the duplicate discard algorithm.

6.1.7 Configuration check

The remaining 4 bits of the RCT carry a distinct identifier for LAN_A or LAN_B, specifically the codes 1010 (“A”) and 1011 (“B”). Therefore, the frames differ in one bit (and in the FCS). The receiver checks that the frame comes from the correct LAN. It does not reject a frame that comes from the wrong LAN, since this could be a legitimate frame which happens to have the length information in its last 12 bits, but it increments the error counters CntErrWrongLanA or CntErrWrongLanB since this could hint at a configuration error. Since this kind of error is permanent, it is detected rapidly.

6.1.8 Network supervision

The health status of each LAN and its attached devices (nodes and switches) is monitored, otherwise redundancy helps little.

The receiver checks that all frames come in sequence and that frames are correctly received over both channels. It maintains error counters that network management can read.

To this effect, all senders and receivers maintain tables of nodes with which they communicate that record the last time a frame was received from another node, the time a multicast or broadcast frame was sent and other protocol information.

At the same time, these tables allow to establish connections to synchronize the sequence numbers and detect sequence gaps and missing nodes.

Since the protocol is loosely connection-oriented, the sequence numbers corresponding to non-existent nodes are cleaned up by a low-priority task after a time NodeForgetTime.

Supervision relies on each DANP sending periodically a PRP_Supervision frame that allows checking the integrity of the network and the presence of the nodes. At the same time, these frames allow checking which devices are DANP, the MAC addresses they use and which operating mode they support, duplicate accept or duplicate discard.

6.1.9 Redundancy management interface

Redundant devices and links are useless without network management supervising this redundancy and calling for maintenance actions.

The LRE presents a network management interface that allows to track the health of each LAN, and especially to detect failures early when the error rate increases. To this effect, the LRE keeps for each adapter (each LAN) a counter of received messages and of messages received with an error.

The LAN statuses appear as SNMPv1 or SNMPv2/v3 variables. This allows using the same tools for managing the nodes and the switches.

NOTE SNMP is part of the IP protocol suite.

6.2 PRP protocol specifications

6.2.1 Installation, configuration and repair guidelines

NOTE These guidelines are to be followed at installation time, they do not apply to conformance testing of the devices.

6.2.1.1 LANs layout

The network shall consist of two LANs that have similar properties, i.e., each one is able to carry the traffic that would exist in the absence of redundancy.

6.2.1.2 Labelling cables

The two LANs shall be labelled A and B and shall use cables distinctly identified.

6.2.1.3 Labelling switches

Switches in the two LANs shall have a distinct label or colour for each A or B.

6.2.1.4 Independent operation

The layout of both LANs shall fulfil the assumption of fail-independence.

6.2.1.5 Configuration

All DANPs shall be configured with the same multicast address for PRP_Supervision frames.

All DANPs shall be configured with the same LifeCheckInterval.

6.2.2 MAC addresses

Both adapters A and B of a DANP shall be configured with the same MAC address.

SANs connected to one LAN only shall not have the same MAC address as another node within the whole network (LAN_A plus LAN_B).

If a DANP implements PICS_SUBS, the MAC address shall be the MAC address of adapter A and adapter B may use a different MAC address, which shall be unique within the whole network (LAN_A plus LAN_B).

NOTE Nodes supporting PICS_SUBS are expected to behave as a DANP that has the default MAC address, address substitution is not specified in this International Standard.

6.2.3 Multicast MAC addresses

All nodes in the network shall be configured to operate with the same multicast address for the purpose of network supervision, see 6.2.7.6.

6.2.4 IP addresses

The IP address(es) of any node or switch within the whole network (LAN_A plus LAN_B) shall be unique.

NOTE A device may have several IP addresses.

A DANP shall have the same IP address(es) when seen from either LAN_A or LAN_B.

Switches on LAN_A and LAN_B are considered as SANs and shall have different IP addresses for the purpose of network management.

6.2.5 Nodes

6.2.5.1 Node types

Doubly attached nodes according to the DANP shall have two network adapters (adapter A and adapter B) that have the same abilities, and, in particular, could be used alternatively if only one LAN is connected, adapter A being connected to LAN_A and adapter B to LAN_B.

SAN have only one adapter for the purpose of this protocol and may be attached to either LAN.

SANs that need to communicate with one another shall be attached to the same LAN or to both LANs through a redundancy box.

6.2.5.2 Labelling connectors

This clause applies to a DANP using two LANs of similar nature.

The connectors for each LAN shall be labelled distinctly as A and B.

When connectors are ordered vertically, LAN_A shall be the upper connector and LAN_B the lower connector in its normal position.

When connectors are ordered horizontally, the left connector shall be the LAN_A and the right connector the LAN_B, as seen from the side where the cables or fibres are plugged.

The redundant connectors shall be independently removable and insertable.

6.2.6 Duplicate accept mode

6.2.6.1 Sending

The sender shall send the frame it receives from its upper layers unchanged over both its adapters so that the two frames appear on the respective LANs.

6.2.6.2 Receiving

The receiver shall forward frames received from both adapters to the upper layers.

NOTE This specification is only testable indirectly, by counting the number of frames over the MIB.

6.2.7 Duplicate discard mode

6.2.7.1 Nodes table

A node shall maintain a table with an entry for each node (SAN or DANP) to which it sends a frame, or from which it receives a frame, using the MAC address as a key. The table shall contain the following information for each unicast, multicast or broadcast address sent by that node.

a) SendSeq
   a 16-bit sequence number used by this node for sending to that remote node or multicast or broadcast address (wrapping through zero)

b) ExpectedSeqA and ExpectedSeqB
   for each adapter A and B, a 16-bit sequence number indicating the sequence number used last by the remote node to communicate with this node on that LAN, incremented by one (wrapping through zero)

c) CntErrOutOfSequenceA and CntErrOutOfSequenceB
for each adapter A and B, a 32-bit error counter indicating that a frame from the remote node was not received in sequence over that LAN

d) StartSeqA and StartSeqB
for each adapter A and B, a 16-bit cursor that limits the drop window

e) CntReceivedA and CntReceivedB
for each adapter A and B, a 32-bit counter indicating the number of frames received over the adapter

f) CntErrWrongLanA and CntErrWrongLanB
for each adapter A and B, a 32-bit counter indicating the number of mismatches on each adapter

g) TimeLastSeenA and TimeLastSeenB
for each adapter A and B, a time field indicating when this node received last a frame from the remote node. This field is in some cases updated at sending to keep track of ageing.

h) SanA and SanB
for each adapter A and B, a Boolean indicating that the remote node is probably a SAN and/or that the remote node uses duplicate accept (see 6.2.7.4.2).

NOTE 1 The table contains for each remote node one row for the unicast frames and one row for each multicast or broadcast address that remote node is sending. It contains one row for each unicast, multicast of broadcast address this node is sending.

NOTE 2 Some fields are irrelevant for a SAN.

NOTE 3 This is a conceptual view, distinct tables for destination and source nodes could be implemented.

6.2.7.2 Redundancy control trailer (RCT)

The RCT inserted into each DANP frame shall consist of four octets, structured in the following way (in the order of transmission).

a) A 16-bit sequence number (SequenceNr) transmitted with the most significant 8 bits in the 1st octet, which reflects the counter SendSeq of the nodes table for the destination of the frame (see 6.2.7.1).

b) A 4-bit LAN identifier (Lan) transmitted as the most significant 4 bits of the 3rd octet, which carries the sequence “1010” for LAN_A, respectively the sequence “1011” for LAN_B.

c) A 12-bit LSDU size (LSDU_size) whose most significant 4 bits are transmitted in the least significant 4 bits of the 3rd octet, that indicates the size in octets of the LSDU starting from the end of the protocol type (PT) field as defined in IEEE 802.3 and 802.1Q (octet offset 12-13 without LAN header or 16-17 with VLAN header) to the RCT, excluding the PT, and the frame part after the RCT, but including the RCT itself.

NOTE Padding inserted before the RCT is included in the LSDU size, padding inserted after the RCT is not included in the LSDU size.

6.2.7.3 Sending (duplicate discard mode)

6.2.7.3.1 Frame size control

The sender shall have the ability to limit the LSDU size so that the complete frame, including the four-octet RCT, does not exceed the maximum size allowed on the LAN when it operates in the duplicate discard mode.

NOTE 1 This maximum size is currently 1518 octets for untagged frames according to IEEE 802.3:2005.

NOTE 2 This specification does not apply to the LRE, but to its upper layers.

6.2.7.3.2 Sending and nodes table

When sending a frame coming from its upper layers, a node shall

a) update the nodes table:

- If the destination address (single cast, multicast or broadcast address) is not yet in the nodes table, create an entry in that table and record as TimeLastSeenA and TimeLastSeenB the current time. If the destination is a unicast address, set the SanA and the SanB to 1, if it is a multicast or broadcast address set them to 0. All other values shall be reset to 0, except for the sequence number SendSeq that may take an arbitrary value, preferably the value 1.
- If the destination address (single cast, multicast or broadcast address) is already in the nodes table, increment the sequence number SendSeq for that address, wrapping over through 0.
- If the destination address is a multicast address or the broadcast address, update in addition the TimeLastSeenA and TimeLastSeenB counters.

NOTE 1 Updating TimeLastSeenA, respectively TimeLastSeenB at sending initializes the ageing time for the remote node. The receiving process actualizes this time value when it receives a frame from that node. A time-out process removes the entry.

NOTE 2 Duplicate discard is assumed for multicast/broadcast addresses, since no PRP_Supervision frame tells the mode. For unicast addresses, the remote node is likely a SAN on LAN_A or LAN_B. If the destination is a DANP, an entry in the nodes table probably exists due to a previously received PRP_Supervision frame, or one is coming soon.

b) send:

- If either SanA or SanB is set, send the frame unchanged over the corresponding adapter.
- If both are set, send the frame unchanged over both adapters.
- If none is set, append the RCT between the LSDU (payload) and before the FCS, preferably just before the FCS if padding is used and send the appended frame with LAN identifier “A” through its adapter A and the frame with LAN identifier “B” through its adapter B, in the same conditions as 6.2.6.1.

6.2.7.4 Receiver (duplicate discard mode)

6.2.7.4.1 Receiving and nodes table

On reception of a frame that is not a BPDU according to IEEE 802.1Q over either adapter, a node shall

a) if the adapter signals that the frame is in error, increment the error counter of the respective adapter CntErrorsA or CntErrorsB and ignore the frame;

b) otherwise,

- if this frame is not a PRP_Supervision frame and not a BPDU and its source is not yet in the nodes table, create an entry in the nodes table for that source MAC address assuming it is a SANA or a SANB, depending which LAN the frame arrives on;
- if the frame is received from LAN_B from a node registered as SANA, or over LAN_A from a node registered as SANB, set SanA = SanB = 1 for that source;
- if this frame is a PRP_Supervision frame, and its source is not yet in the nodes table, create an entry in the nodes table for that source assuming DANP duplicate accept or duplicate discard according to the PRP_Supervision frame contents. If the source is already in the nodes table, update its status to DANP duplicate accept or duplicate discard;
- record the local time at which the frame was received in the TimeLastSeenA, respectively TimeLastSeenB fields of the nodes table for that source;
- increment by one (wrapping through 0) the counters CntReceivedA or CntReceivedB of the nodes table for that source and address kind.

NOTE Updating SanA and SanB allows to move an SAN from LAN_A to LAN_B and vice-versa. If this happens, the DANP will send on both LANs and after NodeForgetTime it will send only on the correct LAN.

6.2.7.4.2 Identification of frames associated with the duplicate discard mode

A receiver shall identify as a duplicate candidate a frame whose last 12 bits before the FCS match the physical size of the LSDU as defined in 6.2.7.2, except for small frames that use padding, for which the receiver shall scan the frame backwards until it finds a matching size field, stopping when reaching the LT field.

NOTE 1 Small frames using padding are smaller than 64 octets.

NOTE 2 Reception of an RCT is not a sufficient criterion to declare its source as DANP, since some protocols reply with the same frame as received.

6.2.7.4.3 LAN identification

A receiver shall check for a frame identified as a duplicate candidate that the four bits previous to the size are either 1010 (A) or 1011 (B)

A receiver shall increment the CntErrWrongLanA, respective CntErrWrongLanB counter of the source device in the Nodes Table if the LAN identifier does not match the adapter from which it received the frame and forward the unchanged frame to its upper layers.

NOTE If one SAN is moved from LAN_A to LAN_B, it will first be considered DANP Duplicate Accept for the duration of NodeForgetTime before it becomes a SAN B.

6.2.7.4.4 Drop window

A receiver shall consider for each LAN and source node the drop window as the range of sequence numbers from StartSeqA to (excluded) ExpectedSeqA, respectively StartSeqB to (excluded) ExpectedSeqB, in Modulo 16 arithmetic.

6.2.7.4.5 Sequence check

The receiver shall check if the received frame is in sequence by comparing it with the ExpectedSeqA, respectively ExpectedSeqB of the LAN over which it was received, and increment the error counter CntErrOutOfSequenceA, respectively CntErrOutOfSequenceB if they are not equal, and then increment ExpectedSeqA, respectively ExpectedSeqB.

6.2.7.4.6 Frame discard

If the sequence number of a frame that is a duplicate candidate is within the drop window of the other LAN, the receiver shall discard that frame, reset the drop window of the LAN over which the frame was received to 0 (StartSeqA:= ExpectedSeqB respectively StartSeqB:= ExpectedSeqA) and move the lower bound of the drop window on the other LAN to one position ahead of the received frame (StartSeqA:= StartSeqB).

6.2.7.4.7 Frame keeping

If the sequence number of a frame that is a duplicate candidate is outside the drop window of the other LAN, the receiver shall forward the frame to the upper layers.

If the sequence number is in sequence on that LAN, the receiver shall, if the maximum window size DropWindowMax (see 6.2.7.8) has been reached, increase by one the lower drop window bound for the LAN over which the frame was received, StartSeqA or StartSeqB.

If the received sequence number is out of sequence, the receiver shall reset the drop window on that LAN to one (for example, StartSeqB:= CurrentSeqB).

6.2.7.4.8 Transparent reception

If the configuration setting TransparentReception of the node is set, the receiver shall not remove the RCT before transferring the frame to the upper layers.

If the configuration setting TransparentReception of the node is not set, the receiver shall remove the RCT on frames where it has identified the presence of the RCT.

6.2.7.5 Clean-up of the nodes table

A node shall clear a nodes table entry when the time elapsed since reception of a frame from that source over both TimeLastSeenA and TimeLastSeenB exceeds NodeForgetTime (see 6.2.7.8).

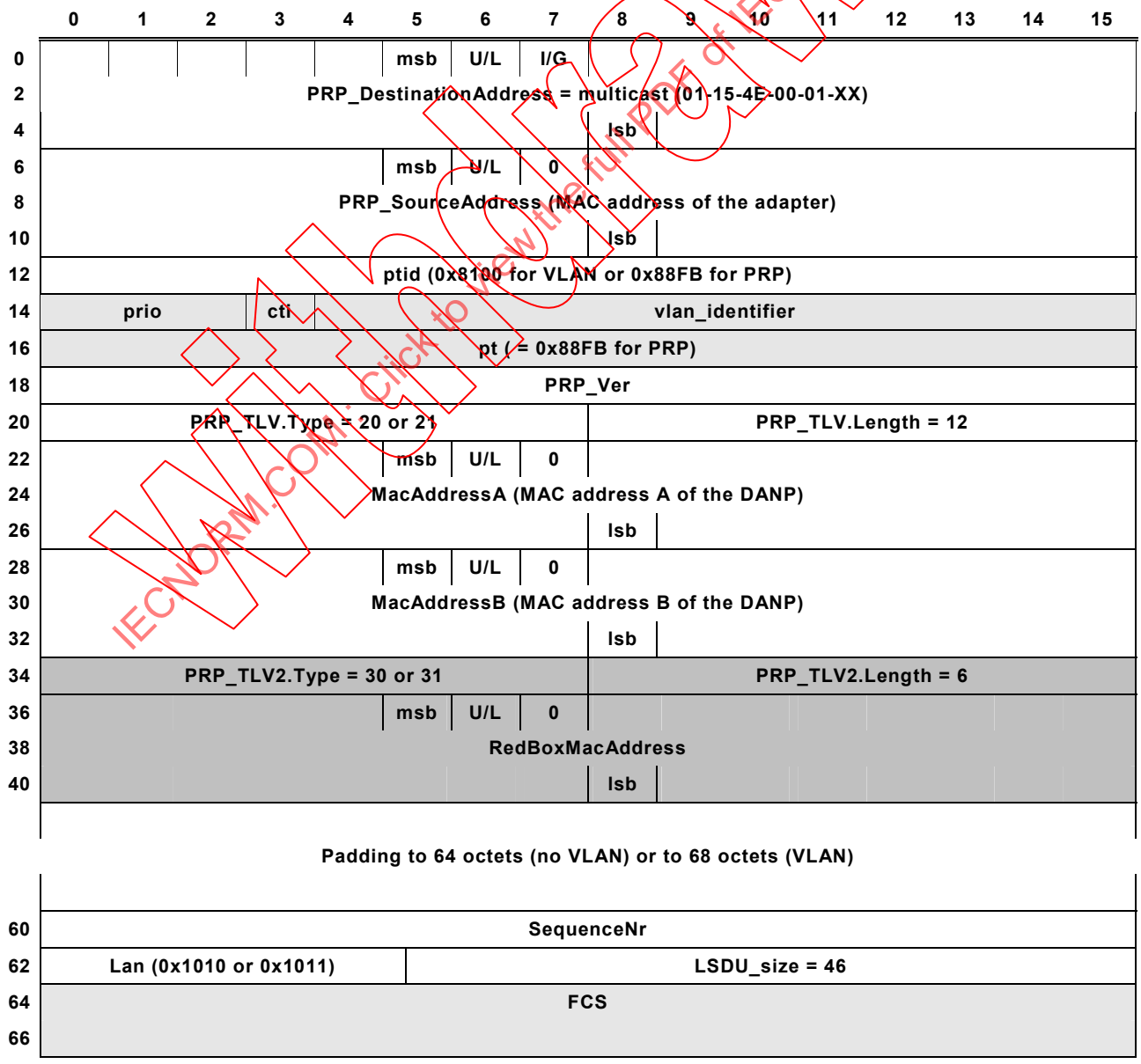
NOTE It is sufficient to check the whole Nodes Table every NodeForgetTime for stale entries.

6.2.7.6 PRP_Supervision frame

6.2.7.6.1 Sending

Each DANP shall multicast a PRP_Supervision frame over both its adapters with the format specified in Table 37 every LifeCheckInterval (see 6.2.7.8). This format shall also be used when the node is operating in duplicate accept mode.

Table 37 – PRP_Supervision frame with VLAN tagging



6.2.7.6.2 PRP_Supervision frame contents

PRP_DestinationAddress
Reserved multicast address 01-15-4E-00-01-XX shall be used for this protocol. By default XX is “00”, but if conflicts arise, XX can be configured to take any value between 0x00 and 0xFF.

PRP_SourceAddress
MAC address of the sending adapter.

PRP_Ver
Indicates the protocol version, set to “0” (zero) for this version of PRP.
Implementation of version X of the protocol shall interpret version >X as if they were version X, ignoring any parameters and/or flags added by the more recent version, and interpret version <=X PRP_Supervision frames exactly as specified for the version concerned.

PRP_TLV.Type
Indicates the operation mode and shall have a value of 20 to indicate that the node supports the duplicate discard or a value of 21 to indicate that it implements duplicate accept. Other values are reserved.

PRP_TLV.Length
Indicates the length of the following MAC addresses (12).

MacAddressA and MacAddressB
MAC addresses used by each port. These addresses shall be identical except if address substitution (PICS=PRP_SUBS) is supported by the sender.

PRP_TLV2
This field shall be set to 0 if the source node is not a redundancy box (see 6.2.7.6.3)

SequenceNr
Sequence number used for PRP_Supervision frames

Lan
LAN over which this PRP_Supervision frame is sent.

LSDU_size
Size of the LSDU, always 46 (independently if tagging is used or not).

The following fields are only sent by a redundancy box when it relays frames on behalf of a SAN and at least the next two octets shall be 0 for other nodes.

NOTE 1 Octets with offset 14 to 17 are inserted only if VLAN according to 802.1D is used.

NOTE 2 The frame has a size of 68 octets if tagging is used to avoid padding if a switch removes the tag.

6.2.7.6.3 PRP_Supervision frame for redundancy box

A redundancy box, i.e., a node acting as a proxy for one or several SANs (called VDAN or virtual DAN) shall append to the TLV field a second TLV field with the following contents:

PRP_TLV2.Type
Indicates the operation mode and shall have a value of 30 to indicate that the node is a redundancy box or a value of 31 to indicate that it is a VDAN. Other values are reserved. This field shall only be sent by a redundancy box, otherwise it shall be zero.

PRP_TLV2.Length
Indicates the length of the following MAC address (6 for a redundancy box, 0 otherwise).

RedBoxMacAddress
MAC address of the redundancy box that acts as proxy for the other device. This field shall only be sent by a redundancy box, otherwise it shall be zero.

6.2.7.6.4 Reception of a PRP_Supervision frame

When receiving a PRP_Supervision frame over any LAN, a node shall create an entry in the nodes table corresponding to the MacAddressA of that source as indicated in the message

body, not in the source address, with the duplicate accept or duplicate discard mode as indicated in the frame.

If MacAddressA and MacAddressB are different, this indicates that the sending node supports PICS_SUBS. If the receiving node supports PICS_SUBS, a receiving node shall, in all frames it receives from that node over adapter A respective adapter B, substitute the received MAC address by the default MAC address of that node (which may be identical to MacAddressA) before forwarding the frame to the upper layers.

6.2.7.6.5 Non-reception of a PRP_Supervision frame

If a node ceases to receive PRP_Supervision frames from a source for a time longer than NodeForgetTime, but receives frames from that source over one LAN only, it shall change the status of this node to SANA, respective SANB, depending on the LAN from which frames are received.

NOTE 1 This rule allows moving a SAN between LAN_A and LAN_B, and also to obtain the right mode for a SAN if it was first registered at sending and not at receiving, since a DANP starts by sending on both LANs.

NOTE 2 This rule allows distinguishing an SAN from a DANP in duplicate accept mode with one line disconnected.

6.2.7.7 Switching end node

If this setting is enabled, the node shall act as a switching end node for its two ports, implementing either:

- SRP (serial redundancy protocol), a subset of IEEE802.1D, Clause 8, in which its ports may only have the root or alternate/backup role, subject to PICS_SRP; or
- RSTP, (rapid spanning tree protocol), the IEEE802.1D, Clause 8, in which its ports can take the root, alternate/backup or designated role, subject to the PICS PRP_RSTP; or
- MRP, see Clause 5, subject to the PICS PRP_MRP.

NOTE 1 The switching end node setting supports attachment of a DANP to two switches of the same LAN to implement a partial redundancy topology. Activating this setting implies duplicate accept. There is no requirement that normal frames should be bridged in case of a double failure, but implementers are free to include this feature.

NOTE 2 No RCT is appended when one of these modes is enabled.

6.2.7.8 Constants

The constant parameters are shown in Table 38.

NOTE Other values may be defined at the user's responsibility.

Table 38 – PRP constants

Constant	Description	Default value
LifeCheckInterval	How often a node sends a PRP_Supervision frame	2 000 ms
NodeForgetTime	Time after which a node entry is cleared	60 000 ms
DropWindowMax	Max size of drop window	32 768

6.3 PRP service specification

6.3.1 Arguments

These arguments are used in both the command and the response. In a command (PRP write), they indicate the desired setting and in a status (PRP read), they indicate the actual setting.

Table 39 – PRP arguments

Argument	Definition	Data type
Node	Node name in the LRE	VisibleString32
Manufacturer	Name of the LRE manufacturer	VisibleString255 (can be read only)
Version	Version of the LRE software	VisibleString32
MacAddressA	MAC address to be used by network interface A	Unsigned48
MacAddressB	MAC address to be used by network interface B	Unsigned48
AdapterActiveA	Adapter A is commanded to be active or responds that it is active if true	Boolean1
AdapterActiveB	Adapter B is commanded to be active or responds that it is active if true	Boolean1
DuplicateDiscard	Duplicate discard algorithm is (to be) used at reception and the RCT is (to be) appended at sending if true	Boolean1
TransparentReception	RCT is not (to be) removed when forwarding to the upper layers if true	Boolean1
SwitchingEndNode	if 0: LRE is not (to be) configured as a switching node if 1: LRE is (to be) configured as an SRP switching node if 2: LRE is (to be) configured as an RSTP switching node if 4: LRE is (to be) configured as an MRP switching node	Integer8
NodesTableClear	Nodes table is (to be) cleared if true	Boolean1
SupervisionAddress	Address to be used for PRP_Supervision frames	Unsigned 48
LifeCheckInterval	Interval at which the PRP_Supervision frame is (to be) sent in milliseconds	Unsigned16
NodeForgetTime	Interval at which the nodes table entry of a node is (to be) cleared, in seconds	Unsigned16
DropWindowMax	Maximum size of the drop window to be used	Unsigned16
CntTotalSentA	Number of frames sent over adapter A	Unsigned32
CntTotalSentB	Number of frames sent over adapter B	Unsigned32
CntTotalReceivedA	Number of frames received over adapter A	Unsigned32
CntTotalReceivedB	Number of frames received over adapter B	Unsigned32
CntErrorsA	Number of transmission errors on adapter A, as signalled by the adapter	Unsigned32
CntErrorsB	Number of transmission errors on adapter B, as signalled by the adapter	Unsigned32
CntNodes	Number of nodes in NodesTable	Unsigned16
NodesTable	Records for all nodes that have been detected within the last NodeForgetTime the following fields	Sequence, see 6.3.2

6.3.2 NodesTable

NOTE 1 The key attribute of the nodes table is MacAddressA as received in the PRP_Supervision frame sent by a DANP.

NOTE 2 Most of these attributes exist not only in one instance per physical remote node, but also as separate instances for each multi/broadcast address used by that node, and some also for each multi/broadcast address used by this (local) node (see 6.2.7.1).

Table 40 – PRP arguments

Argument	Definition	Data type
MacAddressA	MAC address of the source node (6 octets)	OctetString6
MacAddressB	MAC address of the source node (6 octets) as seen over adapter B, as advertised by the PRP_Supervision frame	OctetString6
CntReceivedA	Number of frames received from that source over LAN_A	Unsigned32
CntReceivedB	Number of frames received from that source over LAN_B	Unsigned32
CntKeptFramesA	Number of frames that were kept because they were out of the drop window on LAN_A	Unsigned32
CntKeptFramesB	Number of frames that were kept because they were out of the drop window on LAN_B	Unsigned32
CntErrOutOfSequenceA	Number of frames that were out of sequence on LAN_A	Unsigned32
CntErrOutOfSequenceB	Number of frames that were out of sequence on LAN_B	Unsigned32
CntErrWrongLanA	Number of frames that were received with the wrong LAN identifier on LAN_A	Unsigned32
CntErrWrongLanB	Number of frames that were received with the wrong LAN identifier on LAN_B	Unsigned32
TimeLastSeenA	UTC time at which the latest frame was received over LAN_A.	UTCTime
TimeLastSeenB	UTC time at which the latest frame was received over LAN_B.	UTCTime
SanA	True if the remote device is most probably a SAN accessible over adapter A	Boolean1
SanB	True if the remote device is most probably a SAN accessible over adapter B	Boolean1
SendSeq	Sequence number used to communicate with that remote device	Unsigned16

NOTE MacAddressB is not a key attribute.

6.3.3 PRP Write

This service shall be used to write values to the LRE of a DANP to control the PRP. Table 41 shows the parameters of this service.

Table 41 – PRP Write

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Node	M	M(=)		
Manufacturer	M	M(=)		
Version	M	M(=)		
MacAddressA	M	M(=)		
MacAddressB	M	M(=)		
AdapterActiveA	M	M(=)		
AdapterActiveB	M	M(=)		
DuplicateDiscard	M	M(=)		
TransparentReception	M	M(=)		
SwitchingEndNode	M	M(=)		
NodesTableClear	M	M(=)		
Supervision address	M	M(=)		
LifeCheckInterval	U	U(=)		
NodeForgetTime	U	U(=)		
DropWindowMax	U	U(=)		
Result (+)			S	S(=)
Status			M	M(=)
Result (-)			S	S(=)
Status			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request as defined in 6.3.1.

Result(+)
This parameter indicates that the service request succeeded.

Status
This parameter shall return 0 (no error condition detected)

Result(-)
This parameter indicates that the service request failed

Status
This parameter specifies the error condition (see MIB in Clause C.3)

6.3.4 PRP Read

This service shall be used to read the current status of the LRE from a DANP. Table 42 shows the parameters of this service.

Table 42 – PRP read

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
none				
Result (+)			S	S(=)
Node			M	M(=)
Manufacturer			M	M(=)
Version			M	M(=)
MacAddressA			M	M(=)
MacAddressB			M	M(=)
AdapterActiveA			M	M(=)
AdapterActiveB			M	M(=)
DuplicateDiscard			M	M(=)
TransparentReception			M	M(=)
SwitchingEndNode			M	M(=)
NodesTableClear			M	M(=)
SupervisionAddress			M	M(=)
LifeCheckInterval			M	M(=)
NodeForgetTime			M	M(=)
DropWindowMax			M	M(=)
CntTotalSentA			M	M(=)
CntTotalSentB			M	M(=)
CntErrorsA			M	M(=)
CntErrorsB			M	M(=)
CntNodes			M	M(=)
NodesTable			M	M(=)
Result (-)			S	S(=)
Status			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request as defined in Clause 6.3.1.

Result(+)
This parameter indicates that the service request succeeded.

Result(-)
This parameter indicates that the service request failed.

Status
This parameter specifies the error condition (see MIB in Annex C.3).

6.4 PRP Management Information Base

The MIB objects reflect the arguments of the service parameters which bear the same name, with an uppercase first letter. If the PICS option PRP_MIB is true, the MIB data structures defined in Annex C.3 shall be available at OID = 1.0.62439 in addition to the MIBs that the adapters provide.

6.5 PRP Protocol Implementation Conformance Statement (PICS)

The PICS shall indicate if the following options are supported:

- PRP_MIB: ability to support the SNMP MIB
- PRP_SRP: ability to perform as a reduced RSTP switch element (no designated port role)
- PRP_RSTP: ability to perform as a full RSTP switch element (with designated port role)
- PRP_MRP: ability to perform as an MRP switch element (client or master)
- PRP_SUBS: ability to substitute MAC addresses.

7 CRP – Cross-network Redundancy Protocol

7.1 CRP Overview

This part describes a redundancy protocol executed within the end nodes, as opposed to a redundancy protocol built in the switches. There is no central "Redundancy Manager"; instead each node operates autonomously.

7.2 CRP Nodes

There exist different classes of nodes that may interoperate on the same network:

- DANCs able to execute the CRP protocol, and having two ports for the purpose of redundancy;
- SANCs able to execute the CRP protocol, and having only one port;
- SAN, such as commercially available laptops or file servers that are not aware of the CRP protocol. Even though not aware, SANs can also have access to the redundancy management data for the purpose of monitoring and network management.

In DANCs, these two ports are referred to as port A and port B. They are managed by the LRE, whose implementation is not prescribed, and which is conceptually located in the communication stack below the network layer, as illustrated in Figure 31.

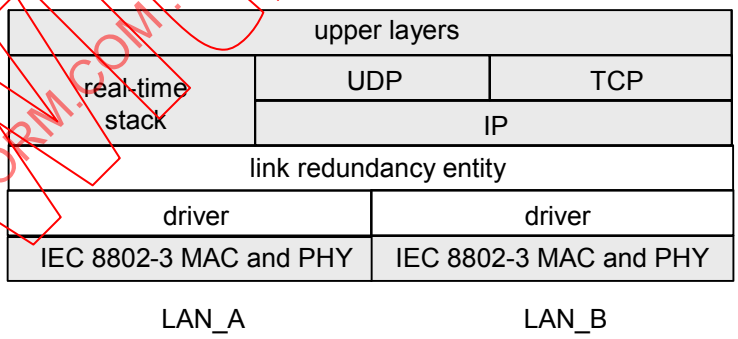


Figure 31 – CRP Stack architecture

This arrangement provides application-level transparency. The LRE hides redundancy from the upper layers and manages the ports. A node can therefore operate with only one IP address.

7.3 CRP LAN topology

Implementing the redundancy protocol within the DANCs allows a variety of topologies, using switches that are not aware of the redundancy protocol and could implement another redundancy protocol such as RSTP.

This standard does not dictate the topology but does allow for configuration of node behaviour to accommodate the characteristics of the specific LAN being used.

Nodes may be attached to the same or to different switches of a single LAN, which may or may not include redundant links, as Figure 32 shows. Attaching both links to the same switch only provides leaf link failure resilience.

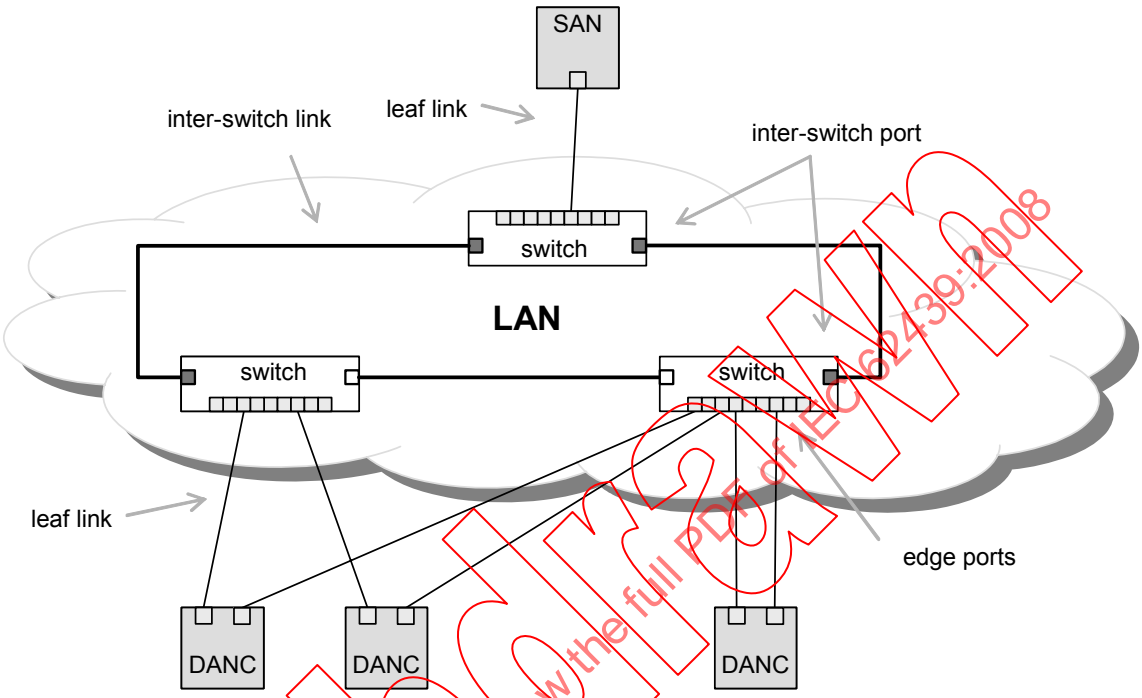


Figure 32 – CRP single LAN topography

Nodes may be attached to separate LANs, which are basically failure-independent, but may be connected by an inter-LAN link, as Figure 33 shows.

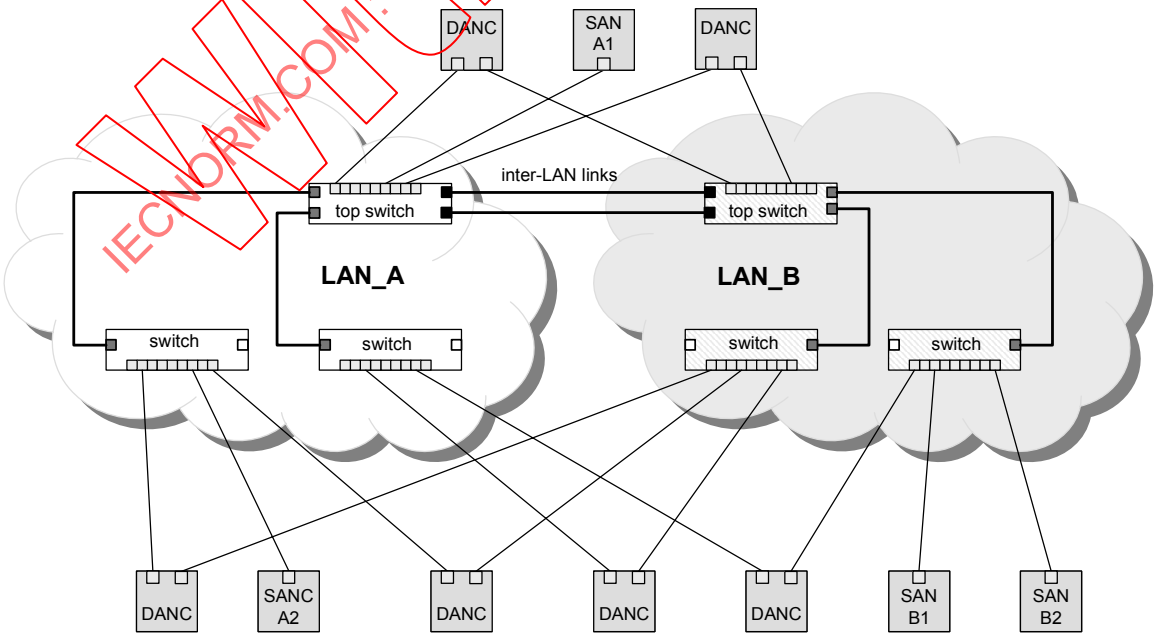


Figure 33 – CRP double LAN topology

When there is only one LAN, a node is attached through both its ports to that LAN. In double LAN configurations, port A is normally connected to LAN_A and port B to LAN_B. Connecting

a node twice to the same network tree or connecting port A to LAN_B and vice versa may be a configuration error called “crossed cables”.

7.4 CRP Key components

7.4.1 CRP General protocol operation

7.4.1.1 Doubly attached nodes (DANCs)

DiagnosticFrames are used to exercise communication paths and to assess the network health. A DiagnosticFrame contains a summary of the reporting node’s view of the network health and status, including its own port.

Annunciation frames are sent to announce the existence of the node. These frames are described in 7.5.7.1

Each DANC sends a pair of DiagnosticFrames periodically, every Tdmi, on both of its ports, as Figure 34 shows. Each DANC that receives one DiagnosticFrame on one port expects the other message of the pair on the other port. (On a single LAN, the node receives both messages on both ports.) If a node receives no message or if it does not receive the second DiagnosticFrame on the other port before receiving several more DiagnosticFrames on the same port, it records a fault in the row of the Network_Status_Table for the corresponding node.

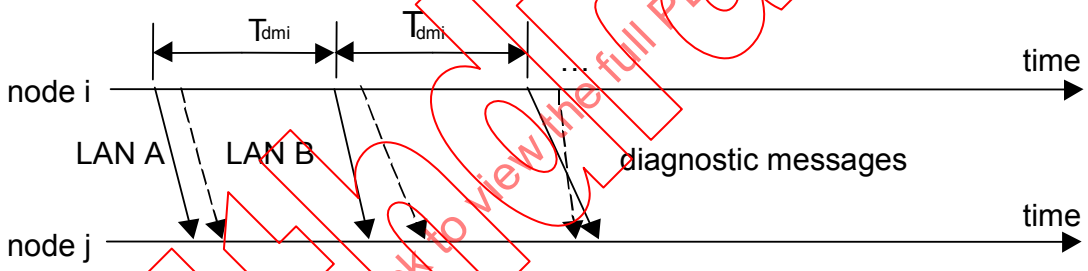


Figure 34 – CRP DiagnosticFrame pair approach

In practice the receiving node compares the Sequence_Number of the last message received on the other port with that just received. If the difference in Sequence_Number is more than the configured Max_Sequence_Number_Difference, a fault is recorded.

Based upon the diagnostic frames it receives from all other nodes, each node can select which port to use to send messages to a particular node, on a node-per-node basis.

EXAMPLE

Figure 35 shows four nodes connected to two redundant LANs which are not connected with each other. Node 3 and 4 have link failures. The diagnostic frame handling on node 3 is detailed.

Each node broadcasts its view on the port status of all nodes it detected in addition to other status information (source MAC address, Node_Index, etc.).

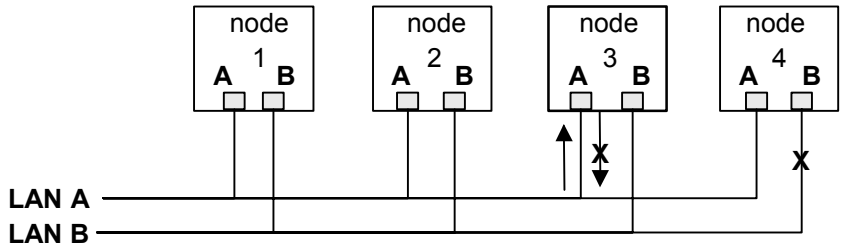
Node 3 maintains a Network_Status_Table populated by the DiagnosticFrames from nodes 1, 2, and 4, as shown in Table 43.

The port status values are OK to indicate a working condition, and X for a don’t know or bad condition.

According to the first three columns of the Network_Status_Table in Table 43, node 3 sends out its Received_DiagnosticFrame for port A as [OK, OK, OK, OK] and for port B as [OK, OK, OK, X].

Similarly, node 1 sends out its view on nodes 2, 3, & 4 as [OK, OK, X, OK] for port A and [OK, OK, OK, X] for port B. Node 3’s adapter A and adapter B status is populated as shown in Table 43.

The row for node 3 is set based on its own testing, but in this example there is no testing, so all appears to be OK.



Node 3: Interface A partial failure; can receive, but not transmit
Node 4: Interface B complete failure.

Figure 35 – CRP Example system

Table 43 – CRP Example Network_Status_Table for node 3

Node_Index #	Received_DiagnosticFrame		Reported status extracted from DiagnosticFrame	
	Received on adapter A	Received on adapter B	Node 3 Received on adapter A	Node 3 Received on adapter B
	Received from adapter A/B [a]	Received from adapter A [a]/B	Received from adapter A/B [a]	Received from adapter A [a]/B
1	OK/X	X/OK	X/X	X/OK
2	OK/X	X/OK	X/X	X/OK
3 (this node)	OK/X	X/OK	OK/X	X/OK
4	OK/X	X/X	X/X	X/X

[a] The cross statuses are all "X" for a dual LAN without inter-LAN link. That is, messages originating from a port A are never heard on a port B and vice versa.

The DiagnosticFrames provide therefore:

- minimal assurance of a working path. With each message received, the receiving node can assume that its own receiver, the reporting node's transmitter, and the path through the network are all working;
- assurance that the reverse path is working. With each message received, the receiving node can extract the reporting node's view of the receiving node and thus determine whether its own transmitter, the reporting node's receiver, and the path through the network are all working.
- This allows the system administrator to construct a variety of coverage strategies such as:
  - ensure that all paths between all nodes are tested;
  - send to a single node. This node may be a "diagnostic node" that only provides detection of faults between each node and the diagnostic node.

7.4.1.2 Singly attached nodes

SANCs also can send and receive DiagnosticFrames. If they choose to transmit them, the DANCs and SANCs are aware of their presence and attempt to ensure that messages reach them. The Network_Status_Table built by the SANC allows it to build DiagnosticFrames and also, in a single LAN, to select a path to a node with a failed port.

7.4.2 CRP Statistics

Statistics should be gathered and presented for each port, by a system management application. Examples of presentation methods are a graphical user interface for visual reporting of network errors or via SNMP.

7.4.3 CRP Network_Status_Table

Each node maintains a Network_Status_Table that holds the node's view of the network.

This table is used to assist with selection of which port(s) to use for transmission to a destination address and which port(s) to use for reception of multicast transmissions.

The Network_Status_Table is constructed from received DiagnosticFrames as well as from other locally acquired and sometimes vendor-specific diagnostic information, for example built-in tests, link integrity pulse, etc.

This table is conceptual and is described to assist with understanding of the concepts in this specification and no specific implementation is prescribed or implied. It is therefore not visible to network management; however,the contents of the table are reflected in the DiagnosticFrames.

The Network_Status_Table in each node keeps for each node and, in some cases, for each port of each note, the following information.

a) For each remote reporting node

- remote node identification (name, index in a table, etc.);
- diagnosis Message interval;
- apparent number of ports (1..N);
- for each port, in nominal order (for example, port to LAN_A before port to LAN_B) the MAC address of the remote port.

b) For each pairing of local and remote ports (for example, A/B, B/A, A/A or B/B)

- time of latest message receipt;
- sequence number of latest received message;
- receipt of DiagnosticFrame from remote port within time;
- remote port's link status from last received DiagnosticFrame.

c) For assessment of remote connectivity

- inferred status of set of ports (functioning properly, cross-connected,...);
- preferred port pairing.

EXAMPLE

Table 44 shows a Network_Status_Table for a SANC and Table 45 a Network_Status_Table for a DANC.

Table 44 – CRP Network_Status_Table for singly connected nodes

Reporting node information			Messages received here sent from reporting node's adapter A				Messages received here sent from reporting node's adapter B			
Node_Index and Node_Name	# of I/Fs	Interval ms	Time	Sequence_ Number of last message received	Diagnostic Frame received	Reported status extracted from DiagnosticFrame for AA	Time	Sequence_ Number of Last message received	Diagnostic Frame received	Reported status extracted from Diagnostic Frame for BA
1 FD001	1	2 000	Time	Number	OK	OK				
2 FD002	1	2 000	Time	Number	OK	OK				
3 LD001	1	1 000					Time	Number	OK	OK
4										
5 LD003	1	5 000	Time	Number	OK	OK				
6	2	5 000	Time	Number	OK	OK	Time	Number	OK	OK
7	1	5 000	Time	Number	OK	OK				

Table 45 – CRP Network_Status_Table for DANC

Reporting node information				Messages received here on adapter A					B (Note 1)		Assessment for reporting node	
				A (Note 2)	Sent from reporting node's adapter B				A (Note3)	B (Note 4)		
Node_Index and name	Addresses for adapters A, B	# Of I/Fs	interval (ms)	…	Time	Sequence_ Number of last message received	Diagnostic Frame received	Reported status extracted from Diagnostic Frame for BA	…	…	Crossed_ cable_status	Selected transmission adapter and receiving adapter
1 FD001	addr1.A, addr1.B	2	2,000	…	Time	Number	OK	OK	…	…	OK	BB
2 FD002	addr2.A, addr2.B	2	2,000	…	Time	Number	OK	OK	…	…	OK	AA
3 LD001	addr3.A, addr3.B	2	1,000	…	Time	Number	OK	OK	…	…	OK	AA
4												
5 LD003	addr5.A, addr5.B	2	5,000	…	Time	Number	OK	OK	…	…	OK	AA
6	addr6.A, addr6.B	2	5,000	…	Time	Number	OK	OK	…	…	OK	BB
7	addr7.A	1	5,000	OK	N/A	N/A	N/A	N/A	NA	NA	OK	AA

NOTE 1 Messages received here on adapter B.

NOTE 2 Sent from reporting node's adapter A.

NOTE 3 Sent from reporting node's adapter A.

NOTE 4 Sent from reporting node's adapter B.

7.4.4 CRP Recovery time

7.4.4.1 Recovery time calculation

The maximum recovery time from a fault is:

$$tr = (1+Max\_Sequence\_Number\_Difference) \times tdmi + tpath + tproc$$

Where tr is the recovery time, tdmi is the time interval of diagnostic frames, tpath is the latency of frame delivery of the network and tproc is the processing time of the receiving LAN redundancy entity.

The value of tpath is determined by the amount of time the packet takes to travel through the network. In a switched network, the worst-case transition time through a FIFO-based switch is dependent on the number of ports of the switch. It may be necessary to use QoS to respect this delay. The arrival of the packet of interest after all other packets bound for the destination interface of interest produces the worst case delay. In this case the delay is:

$$td = Nsp \times tdr \times Sp \times 8$$

where

td is the delay time in a single switch;

Nsp is the number of switch ports on a single switch;

tdr = 1/data rate;

Sp is the maximum packet size in bytes.

EXAMPLE

The following are examples of recovery time for various end node speeds.

Processing time is dominated by the interrupt response time of the end node. For this example an interrupt time of 15 µs is used.

Given: tdmi = 400 ms, tproc = 15 µs, Max_Sequence_Number_Difference = 1, Nsp = 24 and Sp = 1 522 bytes for all cases.

For all end nodes with 10 Mbit/s bandwidth:

$$td = 24 \times 10^{-7} \times 1\,522 \times 8 = 0{,}029 \text{ s}$$

In a single redundant LAN using 6 switches of 24 ports each, and assuming the diagnostic packet traverses all switches, the total tpath is 6 × td = 0,174 s.

Thus,

$$tr = 2 \times 0{,}40 + 0{,}174 + 0{,}000\,015 \text{ s}$$

For this example the maximum processing time is negligible thus:

$$tr = 0{,}974 \text{ s}$$

For all end nodes with 100 Mbps bandwidth, the equation scales directly thus

$$td = 24 \times 10^{-8} \times 1\,522 \times 8 = 0{,}002\,9 \text{ s and for 6 paths} = 0{,}014\,6 \text{ s}$$

Thus

$$tr = 2 \times 0.40 + 0.0174 + 0{,}000\,015$$

$$tr = 0{,}817\,4 \text{ s}$$

7.4.4.2 Maximum repair time

For faults such as cable breaks there is no repair time. However, in a multiple interface switch topology repairing a failed interface requires replacing the entire switch. In this case, powering the switch down to replace it causes additional network disruptions in nodes that have paths though that switch. So as a worst case, the repair time is identical to the recovery time from a fault described in 7.4.4.1.

7.4.5 CRP Multicast messages

7.4.5.1 Sending

Multicast messages are always sent from both ports (if both are operational). They carry as source MAC address the address of the port over which they have been sent.

This applies in particular to the DiagnosticFrames and AnnunciationFrames

7.4.5.2 Receiving

On some network topologies, a single operational multicast message can be received on each of a node's ports. If a node has two ports, the node may be configured to use the Network_Status_Table to select a reception port for multicast operational messages, and so reduce its interrupt and message processing load. Duplicates still need to be detected and discarded.

7.4.6 CRP Unicast messages

7.4.6.1 Sending a frame

Each unicast frame sent by a CRP redundancy participating node is sent from only one port.

When the LRE receives a frame from the IP stack (or other upper layer), it examines the frame, identifies the destination MAC address and looks up for that address in the Network_Status_Table.

If the A-A path to the destination is OK in the table, the LRE sends the frame to port A, which inserts its address as a source MAC address.

If A-A path is NOT OK in the table, but the A-B path is OK, then the LRE substitutes the B MAC address of the destination node in the frame and sends the frame to port A, which inserts its address as a source MAC address.

If the A-A and A-B paths are both NOT OK, but the B-A path is OK, the LRE sends the frame to port B, which inserts its address as a source MAC address.

If the A-A, A-B and B-A paths are NOT OK, the LRE substitutes the B MAC address of the destination node in the frame and send the frame to port B, which inserts its address as a source MAC address.

If the message is broadcast or multicast, the LRE sends the frame to A if A is working and to B if A is not working.

7.4.6.2 Receiving a frame

When the LRE receives a frame from the physical layer over port B, it substitutes the destination MAC B to MAC A address, before forwarding the frame up to the stack. Some IP stacks are sensitive to the IP/MAC association being correct. The IP address is not manipulated/substituted in any way on neither transmit nor receive.

7.4.7 CRP Redundancy information

Each node is configured by a user definable mechanism. The configuration determines the details of how each node transmits DiagnosticFrames and how it uses the Network_Status_Table to select transmission and reception ports. This information can be obtained from another node by listening to AnnunciationFrames.

7.4.8 CRP Redundancy statistics

Any node (even if it is not CRP enabled) may gather and display the health of the nodes in the network by subscribing to the multicast address used for the DiagnosticFrames. The application may then generate a Network_Status_Table and use it in any way.

7.5 CRP Protocol

7.5.1 CRP Singly attached node

SANCs shall have one port and shall participate in the CRP redundancy protocol.

7.5.2 CRP Doubly attached node

DANCs shall have two ports and participate in the CRP redundancy protocol.

7.5.3 CRP Installation, configuration and repair

DANC shall be connected to a single redundant LAN with redundant leaves or to a redundant LAN without redundant leaves as described in 7.3.

NOTE The former connection provides four possible paths to other doubly connected nodes while the latter provides only two.

In order to achieve switch and leaf link redundancy, each port of a DANC shall be connected to a different switch.

SANC and SAN may be connected to any LAN, but preferably all SANCs and SAN shall be connected to the same LAN.

The assigned Node_Index and Node_Name shall be unique in the network.

The maximum Node_Index is 2048; the value of 0 shall not be used for Node_Index.

The maximum number of nodes is 2047.

NOTE 1 This number is limited by the size of the array of status available in the diagnostic packet.

The maximum number of network switch layers of a single redundant LAN with redundant leaves shall be 3.

NOTE 2 This limitation maintains a spanning tree diameter of 7 hops for those networks where a spanning tree is used to prevent loops.

7.5.4 CRP LRE model attributes

7.5.4.1 Attribute specification

7.5.4.1.1 Protocol_Version

This configured attribute specifies the CRP protocol used. It is an Unsigned8 with a value of 0x01.

Packets with higher version numbers shall be rejected by lower versions. Newer versions shall revert to compatibility mode when older version packets are received.

7.5.4.1.2 Number_of_ports

This configured attribute specifies the number of ports on this node for the purpose of redundancy (Unsigned8 = 1 or 2).

7.5.4.1.3 Max_Sequence_Number_Difference

This configured attribute specifies the maximum acceptable difference between the Sequence_Number parameters in a pair of DiagnosticFrame received from a particular sending node. It shall be an Unsigned8 greater than or equal to 1.

NOTE This attribute affects the speed and accuracy of the dual message approach. The value of Max_Sequence_Number_Difference number is at least one to ensure that faults are not detected by normal incrementing of the Sequence_Number. A number of at least two ensures that a single lost message does not cause detection of faults. Having larger numbers allows tolerance of the loss of several successive messages but slows down speed with which the algorithm detects actual faults, as a trade-off between transient tolerance and detection speed.

7.5.4.1.4 Redundancy_Flags

This configured attribute specifies five flags that are not transmitted, indicating one or more of the following.

a) Single multicast message transmission port enabled. This defines the transmission policy for all services with multicast destination addresses except for the DiagnosticFrame.
   – False Transmit on both ports
   – True Transmit on one port
b) Crossed cable handling enabled.
   – False Do not detect crossed cables
   – True Detect crossed cables
c) Single port multicast message reception enabled. This defines the reception policy for all multicast frames except for the DiagnosticFrame_Addresses for ports A and B.
   – False Listen for multicast addresses on both ports
   – True Listen for multicast addresses on one port except if a fault is detected
d) Diagnosis using own messages enabled.
   – False Do not use own DiagnosticFrames for diagnosis
   – True Use own DiagnosticFrames for diagnosis
e) Load balancing enabled.
   – False Do not balance load
   – True Balance load

7.5.4.1.5 DiagnosticFrame_Interval

This configured attribute specifies the time interval in milliseconds between successive sending of the DiagnosticFrames as an Unsigned32.

NOTE This parameter is set individually for each end device. This allows tuning of the intervals to provide fast detection of critical end devices while reducing the traffic from other end devices.

7.5.4.1.6 Ageing time

This configured attribute specifies the time interval in milliseconds used by the LRE to remove silent nodes from its Network_Status_Table.

NOTE 1 The configuration ensures that the value of this attribute is larger than the value of the Max_Sequence_Number_Difference attribute multiplied by the largest value of the DiagnosticFrame_Interval attribute found in received DiagnosticFrames.

NOTE 2 Short ageing times (minutes) mean that DANCs appear and disappear if their power fails briefly. Longer ageing times (days) mean that DANCs that have been removed continue to appear in the Network_Status_Tables.

7.5.4.1.7 DiagnosticFrame_Address

This configured attribute specifies the 32-bit IP address for DiagnosticFrames. The address may be a broadcast or multicast IP address.

The address 224.0.0.105 has been registered for this purpose.

7.5.4.1.8 DiagnosticFrame_UDP_source_port

This configured attribute specifies the 16-bit UDP source port used for DiagnosticFrames.

The same port number as for AnnunciationFrames shall not be used.

7.5.4.1.9 DiagnosticFrame_UDP_destination_port

This configured attribute specifies the 16-bit UDP destination port used for DiagnosticFrames.

The same port number as for AnnunciationFrames shall not be used.

7.5.4.1.10 Annunciation_UPD_port

This fixed attribute specifies the 16-bit UDP destination source and destination port used for AnnunciationFrames as an Unsigned16 with a value of 1 089.

7.5.4.1.11 Node_Index

This configured attribute specifies the 16-bit Node_Index of that node.

NOTE The Node_Index is unique for each node in the scope of the DiagnosticFrame_Address.

7.5.4.1.12 Max_Node_Index

This configured attribute specifies the highest expected Node_Index.

NOTE This number is used to determine the length of the Adapter Status Array and the Crossed Cable Array in the DiagnosticFrame as well as the length of the Network_Status_Table in each end device within the network. The value of this parameter affects the size of DiagnosticFrames; therefore having a very large number adversely impacts performance.

It is recommended that the Max_Node_Index be set to the same value for all devices within the network.

7.5.4.1.13 Node_Name

This configured attribute is a VisibleString32 octets that specifies a Node_Name that is unique for each node in the scope of the DiagnosticFrame_Address. Unused octets in this field shall be transmitted as ASCII space characters.

7.5.4.1.14 Annunciation_Interval

This configured attribute specifies the time interval at which AnnunciationFrames are sent.

7.5.4.1.15 Operational IP Address

This configured attribute specifies the IP address used for application communication.

7.5.4.1.16 Duplicate detection state

This dynamic attribute indicates possible detections of multiple addresses.

It shall be reset to all 0 by configuration or start-up and set upon detection of a conflicting network situation.

It is a bit set encoded as:

0	1	2	3	4	5	6	7
RES						DND	DID

Bit 0 — 5: Reserved
set to 0

Bit 1: DND
1: duplicate name detected,

Bit 0: DID
1: duplicate node index detected,

7.5.4.1.17 LRE state

This dynamic attribute specifies the state of the LRE as one octet.

0	1	2	3	4	5	6	7
CONF							SYN

Bit 0 — 6: CONF
0 = reserved
1 = no name configured
2 = operational
3 – 127 = reserved

Bit 7: SYN
0 = Not synchronized with time server
1 = Synchronized with time server

The LRE state shall be reset to all “0” by configuration or start-up.

7.5.4.1.18 Sequence_Number

This dynamic attribute is a monotonically increasing 32-bit integer that shall start with 0 for the first DiagnosticFrame sent after start-up, incrementing by 1, and rolling over to 0.

7.5.4.1.19 Path status

This dynamic attribute summarizes the view of the node on its paths. It consists of four path status sets as shown in Table 46.

Table 46 – CRP Path_Status_Sets

	Remote node sent over	This node received over
Path_Status_A_to_A	A	A
Path_Status_B_to_A	B	A
Path_Status_A_to_B	A	B
Path_Status_B_to_B	B	B

Each path status set consists of a 32-bit aligned sequence of bits, each bit representing a node, using the Node_Index as an offset and indicating the status of the transmission path from the reporting node to this node.

The following values are used for each bit:

- 0 = DiagnosticFrames successfully received
- 1 = DiagnosticFrames not received

EXAMPLE

Table 47 shows a Path_Status_Set for the node with Node_Index 4. This node (4) indicates that it sees:

Node 1: received “A” messages over its LAN_A, and “B” messages over its LAN_B

Node 2: did not receive messages for a time longer than Aging_Time

Node 3: received only “A” messages over its LAN_A (it’s possibly a SAN)

Node 4: received its own messages over the same port they were sent

Node 5: received “A” messages over its LAN_A, and “B” messages also over its LAN_A – possible configuration error.

Node 6: received “A” messages over its LAN_B, and “B” messages also over its LAN_A – possible crossed cable error.

Table 47 – CRP Example of a Path_Status_Set

0		2		4		6		8		10		12		14		16		18		20		22		24		26		28		30	
Path_Status_A_to_A																															
0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Path_Status_B_to_A																															
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Path_Status_A_to_B																															
0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Path_Status_B_to_B																															
0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.5.4.1.20 Configuration_Version

This attribute specifies the version of this object as an Unsigned16. Each time the value of any of its other attributes changes by configuration, the version number shall be incremented by 1. Version number 0 indicates that this object has not been configured. This value shall be skipped when rolling over.

7.5.4.2 Impact of LRE configuration attributes

The impact of the LRE configured attributes is summarized in Table 48.

Table 48 – CRP configuration attributes impact on LAN operation

Configuration parameter	Dual LAN	Single LAN
Node_Index	Same for both, see 7.5.4.1.10	
Max_Node_Index	Same for both, see 7.5.4.1.12	
Max_Sequence_Number_Difference	Same for both, see 7.5.4.1.3	
DiagnosticFrame_UDP_destination_port	Same for both, see 7.5.4.1.9	
Redundancy_Flags	This parameter consists of the following five flags	
Single Multicast Message Transmission Adapter Enabled	Normally false, allowing multicast messages to be sent on both LANs if the error conditions and presence of SANCs warrants it. Where receiving node loading is critical, however, this may be set true	Normally set true, allowing multicast messages to propagate to both adapters of other nodes. Sending on both adapters increases traffic on the LAN and the receiving nodes
Crossed_cable_detection_enabled	Normally set true. This allows detection of crossed cables	Normally false. Crossed cables have irrelevant on a single LAN
Single Multicast Message Reception Adapter Enabled	Normally false, allowing reception of multicasts on either adapter. Because each multicasting node can choose which to send on, listening on both adapters is essential to hear them	Normally true, allowing operating multicast messages propagate to both adapters of other nodes. Listening on both adapters increases traffic on the receiving node
Diagnosis Using Own Messages Enabled	Normally false, preventing DiagnosticFrame sent on one adapter will not be heard on the other	Normally true, allowing DiagnosticFrames sent from one adapter to be received on the other. Using this as a diagnostic improves fault detection, at the cost of processor load
Load Balancing Enabled	May be set false on a temporary basis to facilitate system maintenance	Normally set false. Load balancing is not applicable to single LAN networks
DiagnosticFrame_Interval	Same for both, see 7.5.4.1.5	
Aging_Time	Same for both, see 7.5.4.1.6	
DiagnosticFrame Adapter A Send Address	The address to which DiagnosticFrames are sent on adapters A and B. Usually the same address is used for both adapters	The addresses to which DiagnosticFrames are sent on adapters A and B. The same address may be used to maximize the ability to detect and recover from faults, at a cost of CPU load. Different addresses allow a reduction of traffic seen by the receiving node
DiagnosticFrame Adapter B Send Address		
DiagnosticFrame Adapter A Receive Address	The address to which DiagnosticFrames are received on adapters A and B. Usually the same address is used for both adapters	The addresses to which DiagnosticFrames are received on adapters A and B. The same address may be used to maximize the ability to detect and recover from faults, at a cost of CPU load

7.5.5 CRP Encoding of the DiagnosticFrame

A DiagnosticFrame shall have the content described in Table 49.

Table 49 – CRP DiagnosticFrame format

Parameter name	Offset	Data type	Size	Description
Ethernet DLL header				
Preamble	0		8	Alternating ones and zeros
Destination address	8		6	Broadcast, multicast MAC address
Source address	14		6	Unicast source MAC address of adapter used to send this message.
Type	20		2	0x800 for IP datagrams
IP header				
Version	22		4	IP Version (4 bit field) = 4
Internet header length				Internet Header Length (4 bit field) is the length of the internet header in 32 bit words = 6
Type of service				Type of service (8 bit field) – set all fields = 0 Bits 0-2: Precedence. Bit 3: 0 = Normal Delay, 1 = Low Delay. Bits 4: 0 = Normal Throughput, 1 = High Throughput. Bits 5: 0 = Normal Reliability, 1 = High Reliability. Bit 6-7: Reserved for Future Use.
Total length				Length of IP field (16 bit field) in bytes, including IP header and data = 346
Identifier	26		4	Identifier for fragmented packets. Set = 0
Flags				Flags (3 bit field) Bit 0: reserved, shall be zero Bit 1: Don't Fragment = 1 Bit 2: 0 = Last Fragment
Fragment offset				Position in original Datagram. First fragment = 0
Time to live	30	Unsigned8	4	Set to 1
Protocol		Unsigned8		User Datagram = 17 (decimal)
Header checksum				16 bits. The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.
Source address	34		4	Source IP address.
Destination address	38		4	Configured DiagnosticFrame_Address, see 7.5.4.1.7
Options	42		1	
Pad	43		3	Pad header to a 32 bit boundary. Pad set = 0
UDP header				
Source port	46		2	See 7.5.4.1.8
Destination port	48		2	See 7.5.4.1.9
Length	50		2	Length of UDP field
Checksum	52		2	Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data.
Header				
Protocol_Version	54	Unsigned8	1	see 7.5.4.1.1
reserved	58	Unsigned24	3	Set to 0x801001
Auxiliary address	58	Unsigned32	4	Not used – set to 0
Message length	62	Unsigned32	4	Specifies the number of octets contained in the entire message, starting immediately after the UDP header until the end of the message.

Parameter name	Offset	Data type	Size	Description
Body				
Node_Index	66	Unsigned16	2	See 7.5.4.1.10
Number of adapters	68	Unsigned8	1	See 7.5.4.1.2
Transmission adapter	69	Unsigned8	1	Port used to transmit this DiagnosticFrame. 0 = Adapter A 1 = Adapter B
DiagnosticFrame_Interval	70	Unsigned 32	4	See 7.5.4.1.5
Node_Name	74	VisibleString	32	See 7.5.4.1.13
Reserved	106	Unsigned8	1	Reserved, set to zero
Duplicate detection state	107	Unsigned8	1	See 7.5.4.1.15
Number of adapter statuses	108	Unsigned16	2	Number of Unsigned32 entries in the Path_Status.
Path_Status_A_to_A	110	array of Unsigned32	see Note	See 7.5.4.1.19
Path_Status_B_to_A	see Note	array of Unsigned32	see Note	See 7.5.4.1.19
Path_Status_A_to_B	see Note	array of Unsigned32	see Note	See 7.5.4.1.19
Path_Status_B_to_B	see Note	array of Unsigned32	see Note	See 7.5.4.1.19
Sequence_Number	see Note	Unsigned32	4	See 7.5.4.1.18
Ethernet DLL trailer				
FCS	see Note		4	CRC based frame check sequence

NOTE The field size and offset depends on the number of adapter statuses, the size of each Path_Status field is 4 x number of adapter statuses, the offset is incremented by the size of the previous field.

7.5.6 CRP Encoding of the AnnunciationFrame

An AnnunciationFrame shall have the format shown in Table 50.

Table 50 – CRP AnnunciationFrame

Parameter name	Offset	Data type	Octet length	Description
Preamble	0		8	Alternating ones and zeros
Destination address	8		6	Broadcast, multicast MAC address
Source address	14		6	Unicast source MAC address of adapter used to send this message.
Type	20		2	0x800 for IP datagrams
IP header				
Version	0		4	IP Version (4 bit field) = 4
Internet header length				Internet Header Length (4 bit field) is the length of the internet header in 32 bit words = 6
Service				Service (8 bit field) – set all fields = 0 Bits 0-2: Precedence. Bit 3: 0 = Normal Delay, 1 = Low Delay. Bits 4: 0 = Normal Throughput, 1 = High Throughput. Bits 5: 0 = Normal Reliability, 1 = High Reliability. Bit 6-7: Reserved for Future Use.
Total length				Length of IP field (16 bit) in bytes, including IP header and data

Parameter name	Offset	Data type	Octet length	Description
Identifier	4		4	Identifier for fragmented packets. Set = 0
Flags				Flags (3 bit field) Bit 0: reserved, set to zero Bit 1: Don't Fragment = 1 Bit 2: 0 = Last Fragment
Fragment offset				Position in original Datagram. First fragment = 0
Time to live	8		4	8 bits. Set to 1
Protocol				8 bits. User Datagram = 17 (decimal)
Header checksum				16 bits. The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.
Source address	12		4	Source IP address of this node.
Destination address	16		4	see 7.5.4.1.7.
Options	20		1	IP option field
Pad	21		3	Pad header to a 32 bit boundary. Pad set = 0
UDP header				
Source port	0		2	Annunciation port see 7.5.4.1.8
Destination port	2		2	Annunciation port see 7.5.4.1.9
Length	4		2	Length of UDP field
Checksum	6		2	Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data.
Body				
LRE_State	0	Unsigned8	1	
Reserved0	1	Unsigned8	1	Reserved
Reserved1	2	Unsigned8	1	Reserved
Duplicate Detection State	3	Unsigned8	1	see 7.5.4.1.15
Node_Index	4	Unsigned16	2	see 7.5.4.1.10
Max_Node_Index	6	Unsigned16	2	see 7.5.4.1.12
Operational IP Address	8	OctetString	16	see 7.5.4.1.17
ReservedString	24	VisibleString	32	Reserved
Node_Name	56	VisibleString	32	see 7.5.4.1.13.
Annunciation_Interval	88	Unsigned32	4	see 7.5.4.1.14
DiagnosticFrame_UDP_Port	92	Unsigned16	2	see 7.5.4.1.8 and 7.5.4.1.9
Reserved	94	Unsigned16	2	Reserved, set to 0.
Configuration_Versionr	96	Unsigned32	4	see 7.5.4.1.1
User_Option	100	Unsigned32	4	Reserved
Number of Entries	104	Unsigned32	4	indicates the number of user-defined entries
Entries	108	Unsigned32	4 x Number of Entries	Reserved for user-defined entries
Ethernet DLL trailer				

Parameter name	Offset	Data type	Octet length	Description
FCS	0		4	CRC based frame check sequence

7.5.7 CRP Common protocol

7.5.7.1 AnnunciationFrames

7.5.7.1.1 Sending

A node shall send an AnnunciationFrame with the format indicated in Table 50:

- after initial power up or warm start;
- following configuration of the node parameters;
- periodically at a rate determined by the Annunciation_Interval.

A SANC shall send the AnnunciationFrame over its port A.

A DANC shall send the AnnunciationFrame over both its port A and port B.

7.5.7.1.2 Receiving

The receiving node shall include the node sending the AnnunciationFrame to the Network_Status_Table.

7.5.7.2 DiagnosticFrames

7.5.7.2.1 Sending

Nodes shall transmit DiagnosticFrames with the format specified in Table 49:

– at startup, or warm start provided the node is configured;
– following a successful node configuration;
– otherwise, at the rate specified by the DiagnosticFrame_Interval (7.5.4.1.5) attribute;
– as long as the node is in an operational state.

A SANC shall send a DiagnosticFrame over its port A, with

– the value of the number of ports field set to 1;
– the Sequence_Number incremented by 1 for each successive message sent;
– the sending adapter set to “1” (LAN_A).

A DANC shall send a pair of DiagnosticFrame over both its port A and port B, with

– the value of the number of ports field set to 2;
– the Sequence_Number incremented by 1 for each successive sending of a pair, and identical Sequence_Numbers fields in both messages of a pair;
– the associated MAC address set;
– the sending adapter set to the value corresponding to the adapter over which the message is sent;
– a time spacing not greater than 30 % of the configured DiagnosticFrame_Interval attribute between the two messages of a pair.

7.5.7.2.2 Receiving

The node shall be configured to receive DiagnosticFrames on its port using the address specified by the DiagnosticFrame_Address attribute at the DiagnosticFrame_UDP_Destination_Port; see 7.5.4.1.9.

7.5.7.3 Detection of duplicate Node_Index

If a node receives a DiagnosticFrame from another node that has the same Node_Index as its own, the node shall set the Duplicate_Detection_State attribute to indicate a duplicate Node_Index was detected.

This entry shall be cleared by reconfiguration of the node with a non-duplicate Node_Index.

7.5.7.4 Detection of duplicate Node_Name

If a node receives a DiagnosticFrame from another node that has the same Node_Name as its own, the node shall set the Duplicate_Detection_State attribute to indicate a duplicate name detected.

This entry shall be cleared by reconfiguration of the node with a non-duplicate Node_Name.

7.5.7.5 Failure detection based on arrival of DiagnosticFrames

DiagnosticFrames received by a node shall be used in the construction of the Network_Status_Table unless

- they indicate duplicate Node_Index detected;
- they have a Node_Index greater than Max_Node_Index; or
- their Node_Index is the same as that for this node.

The node shall track the messages arriving from each reporting node by their Sequence_Number.

The DiagnosticFrame_Interval shall not be considered when evaluating arriving messages.

Arrival of the message shall mark the receive status entry for that reporting node and path that did not receive the message as OK.

If a DANC receives a DiagnosticFrame with a Sequence_Number that exceeds by Max_Sequence_Number_Difference the last DiagnosticFrame received on any other path from the given reporting node, it shall mark the DiagnosticFrame received entry for that reporting node as not OK.

The status shall be returned to OK following receipt of a DiagnosticFrame from the reporting node on the path for which a failure had been recorded provided that its Sequence_Number is no smaller than Max_Sequence_Number_Difference to the highest number received. This shall be the only condition that causes the status to be reset to OK.

The DANC shall reflect the status array of the reporting node contained in the DiagnosticFrame in its Network_Status_Tables in the entry associated with the Node_Index of the reporting node.

However, when a node does not receive a DiagnosticFrame for a particular Node_Index for a time superior to Aging_Time, the adapter A and adapter B columns in the Network_Status_Table for that Node_Index shall indicate that all paths to that node are not OK.

NOTE In DiagnosticFrames sent by receiving nodes, the list of port status reflects this not OK status for that Node_Index.

When receiving nodes begin receiving DiagnosticFrames from a node with that Node_Index, it shall not update the messages received here sent from reporting node's adapter A and B columns in the Network_Status_Table for that Node_Index until Max_Sequence_Number_Difference messages are received. This ensures the DiagnosticFrame content is set correctly.

Two entries of this array shall be examined: that corresponding to the receiving node and that corresponding to the reporting node; see 7.5.7.6.

7.5.7.6 Status array entries

7.5.7.6.1 Receiving node entry

The list of DiagnosticFrame received status array shall indicate whether the reporting node has successfully received DiagnosticFrames on each of the four possible paths from this node. This node shall copy this information from the reporting node's DiagnosticFrame into all four of the reported status extracted from the DiagnosticFrame columns of its Network_Status_Table.

7.5.7.6.2 Reporting node entry

This entry shall indicate that the reporting node has determined whether there are errors on any of the four possible paths from this node. If this entry indicates any status is not OK, it shall be used to update the DiagnosticFrame received status of the Network_Status_Table for the reporting node. Entries with a status of OK shall not used.

NOTE This may record the fault more quickly than waiting for detection based on the dual message approach.

7.5.7.7 Other failure detection

Nodes should provide for the option to use vendor-specific diagnostics or commonly used mechanisms to update their own row in the Network_Status_Table.

The faults detected this way should be sent with the DiagnosticFrame and should allow propagation of the detected fault. Whether or not a node detects errors in its adapters, it shall ensure that its own row is correctly initialized and maintained with an OK value.

7.5.8 CRP Operational messages

7.5.8.1 Load balancing

Nodes that participate in the CRP redundancy approach are able to balance the load for unicast destination addresses between the available transmission adapters. The Redundancy_Flag load balancing enabled shall control the use of load balancing. Subclause 7.5.8.3 defines how the port is selected when load balancing is used.

7.5.8.2 LAN and port maintenance

Maintenance may be performed on LAN components or port with reduced impact on system operation, if the port or LAN components are not being used. To avoid use of the port or LAN components, the Redundancy_Flag Load Balancing Enabled shall be set to false.

NOTE Setting this flag to false has the effect of reverting to the method of using the path associated with this node's operational IP address, or best path if there is a fault.

7.5.8.3 Selecting transmission path

7.5.8.3.1 General

Selected transmission paths indicate the ports used to transmit operational messages to specific destination addresses. A path shall be defined as the combination of the transmitting node's sending port and the destination address for the receiving node. Path selection shall be evaluated for a specific unicast or multicast destination address at initial use, and then shall be determined following the detection of a fault or the detection of a repaired fault as described in 7.5.8.3.2 and 7.5.8.3.3.

7.5.8.3.2 Unicast destination address

Path for unicast destination addresses shall be selected according to Table 51.

Table 51 – CRP unicast destination address handling

Destination address of:	Path selection
CRP redundancy participating node with DiagnosticFrame Received fields and Reported Status Extracted from DiagnosticFrame fields in Network_Status_Table all OK in the AtoA and BtoB columns. Load Balancing Enabled is true	Select path randomly (see Note) and fairly (equal probability of selection) between the adapters
CRP redundancy participating node with DiagnosticFrame Received fields and Reported Status Extracted from DiagnosticFrame fields in Network_Status_Table all OK. Load Balancing Enabled is false	Select path that is associated with this node's Operational IP Address that was configured.
CRP redundancy participating node with DiagnosticFrame Received fields and Reported Status Extracted from DiagnosticFrame fields in Network_Status_Table that have one or more Not OK	Select path with no fault
CRP redundancy participating node with Number of Adapters = 1 (singly connected node)	Select path that leads to available adapter
Node that does not have a row in the Network_Status_Table, (it is not participating in CRP redundancy)	Use existing path
Nodes that are outside the network, nodes reachable through routers	Use existing path
NOTE "Randomly" means that the node selects for a specific path one adapter rather than the other by using a true random number generator, or a pseudo-random number generator with a different seed, each time the node initializes.	

7.5.8.3.3 Multicast destination address

7.5.8.3.3.1 Single multicast message transmission adapter enabled state is true

The single_multicast_message_transmission attribute applies to operational messages.

If there are no errors, the node should send on the adapter that has the most singly connected nodes recorded in the Network_Status_Table. This should be selected by choosing the adapter that has the least number of not applicable entries in its DiagnosticFrame received columns of the Network_Status_Table.

If there is a single error, the node shall send on the adapter that has no error recorded in its DiagnosticFrame received or nodes reporting problems columns of the Network_Status_Table.

Otherwise if there are multiple errors, the node shall send on the adapter that has fewest errors recorded in its DiagnosticFrame received and nodes reporting problems columns of the Network_Status_Table.

If the number of errors is the same for each adapter, no change should be made to the selected transmission adapter for multicast addresses.

7.5.8.3.3.2 Single_multicast_message_reception_adapter_enabled state is false

If all SANs and DANCs can be reached by sending on a single adapter the node shall send on that adapter. That is, if all the “not OK” and “not applicable” entries recorded in its DiagnosticFrame received and nodes reporting problems columns of the Network_Status_Table are associated with the other port.

Otherwise, the node shall send the multicast over both ports.

7.5.8.4 Selecting reception adapter

NOTE The intention is to reduce duplicate operational multicast messages.

Nodes shall select a reception adapter when the Redundancy_Flags [single_multicast_message_reception_adapter_enabled], is true.

If [single_multicast_message_reception_adapter_enabled] is false, the node shall listen on both adapters. If it is true, the requirements shall be as follows:

If: no LAN faults are detected, the node listens on its port A. (SANs or SANCs only have an Adapter A.)

Else: use same format as above. If a single LAN fault is detected indicating that one adapter of this node cannot correctly receive transmissions, use other adapter of this node.

If there are multiple faults associated with both adapters, listen on both adapters.

7.5.8.5 Crossed_cable_status

DiagnosticFrames include a field that identifies the port (A or B) on which the message was sent. For certain LAN topologies, this can be used to determine if the port is connected to the LAN correctly.

7.5.8.6 Configured parameters

Table 52 contains the minimum parameters that are necessary for the protocol. Additional parameters may be added at the user option.

Table 52 – CRP configuration parameters

parameter	description	data type
Device ID	Descriptive string of the end node	OctetString32
Node_Name	End node name	OctetString32
Node_index	End node’s unique device index	Unsigned16
DiagnosticFrame_UDP_destination_port	UDP port used to receive CRP redundancy messages	Unsigned16
Repeat time	End node’s annunciation message repeat time.	Unsigned32
Max_Node_ Index	Highest device index used in the CRP network	Unsigned16
Operational IP address	End node’s operational IP address	OctetString16

7.5.9 CRP services

7.5.9.1 Configuration options and services

A node may obtain its configuration in three optional ways.

The first is through a local configuration page for those nodes that have a graphical user interface, such as a PC.

The second is through switch configuration where a set of switches configures the device index. The operational IP address would be the sum of a base IP address configured in the device firmware plus the device index.

The third is by a network management node using over the network services that commence upon the reception of an annunciation message. In addition, the network management node has the services to gather the statistics kept by each redundancy capable node. The services are:

- set assignment information;
- get redundancy information;
- set redundancy information;
- get redundancy statistics.

7.5.9.2 LAN redundancy service specification

7.5.9.2.1 Set assignment information service

This confirmed service shall be used to set the CRP attributes using the service parameters shown in Table 53.

Table 53 – CRP set assignment info service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Invoke ID	M	M(=)		
Source IP address		M		
Source Port	M	M(=)		
Destination IP address	M	M(=)		
FDA address	M	M(=)		
Device ID	M	M(=)		
Node_Name	M	M(=)		
Node_index	C	C(=)		
DiagnosticFrame_UDP_destination_port	C	C(=)		
Repeat time	C	C(=)		
Clear duplicate detection state	C	C(=)		
Max_Node_ Index	C	C(=)		
Operational IP Address	C	C(=)		
Result (+)			S	S(=)
Invoke ID			M	M(=)
Source IP address				M
Destination IP address			M	M(=)
Destination port			M	M(=)

Parameter name	Req	Ind	Rsp	Cnf
Repeat Time			C	C(=)
Max_Node_index			C	C(=)
Result (-)			S	S(=)
Invoke ID			M	M(=)
Source IP address				M
Destination IP address			M	M(=)
Destination port			M	M(=)
Error info			M	M(=)

Argument

The argument conveys the parameters of the service request

Invoke ID
This parameter contains a value that is determined by the generator of the request and is matched by the responder
(Unsigned32)

Source IP address
This parameter is the IP address from which the service request was sent. The responder uses it when returning the response.
(OctetString16)

Source Port
This parameter is the UDP port from which the service request was sent. The responder uses it when returning the response.
(Unsigned16)

Destination IP address
This parameter is the IP address to which the service request is to be sent. The responder uses it when returning the response.
(OctetString16)

FDA address
This parameter contains the address to which the service request is being sent.
(Unsigned32)

Device ID
This parameter contains a descriptive string of the end node.
(OctetString32)

Node_Name
This parameter contains the value of the end node name. The value is not permitted to be blank.
(OctetString32)

Node_index
This conditional parameter contains the value of the end node's unique device index. Its value is not permitted to be zero.
(Unsigned16)

DiagnosticFrame_UDP_destination_port
This conditional parameter contains the value of the UDP port used to receive CRP redundancy messages.
(Unsigned16)

Repeat time
This conditional negotiable parameter contains the value of the end node's annunciation message repeat time.
(Unsigned32)

Clear duplicate detection state
This conditional parameter causes the duplicate detection state to be set to no duplicates detected if it contains a non-zero value.
(Unsigned8)

Max_Node_Index
This conditional negotiable parameter contains the value of the highest device index used in the CRP network.
(Unsigned16)

Operational IP Address
This conditional parameter contains the end node's operational IP address.
(OctetString16)

Result (+)

This parameter indicates that the service request succeeded. The following fields are included in the response. Negotiable parameters may be different then the ones sent by the network manager.

Invoke ID
Source IP address
Destination IP address
Destination port
Repeat time
Max-Node_index

Result (-)

This parameter indicates that the service request failed. The following fields are included in the response.

Invoke ID
Source IP address
Destination IP address
Destination port
Error info
This parameter specifies the error condition.

7.5.9.2.2 Get redundancy info service

This confirmed service shall be used to retrieve CRP Redundancy attributes using the service parameters shown in Table 54.

Table 54 – CRP get redundancy info service

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Invoke ID	M	M(=)		
Source address		M		
Destination address	M	M(=)		
Result (+)			S	S(=)
Invoke ID			M	M(=)
Source address				M
Destination address			M	M(=)
Redundancy attributes version			M	M(=)
Number of network adapters			M	M(=)
Max message number difference			M	M(=)
Redundancy flags			M	M(=)

Parameter name	Req	Ind	Rsp	Cnf
Diagnostic message interval			M	M(=)
Aging time			M	M(=)
DiagnosticFrame send adapter addresses			M	M(=)
DiagnosticFrame receive adapter addresses			M	M(=)
Result (-)			S	S(=)
Invoke ID			M	M(=)
Source address				M
Destination address			M	M(=)
Error info			M	M(=)

Argument

The argument conveys the parameters of the service request:

Invoke ID
Source IP address
Destination IP address

Result(+)

This selection type parameter indicates that the service request succeeded. The following fields are included in the response.

Invoke ID
Source IP address
Destination IP address
LAN redundancy attributes version
This attribute specifies the version of this object. Each time the value of any of its other attributes changes, the version number is incremented by 1. Version number 0 indicates that this object has not been configured.
(Unsigned32)

Number of network adapters
This attribute specifies the number of network interfaces on this device. A device may have one or two network interfaces. They are labelled Network Interface A and Network Interface B (Network Interface B is only used if there are two network interfaces)
(Unsigned8)

Max message number difference
This attribute defines the maximum acceptable difference between the message number parameters in a pair of Diagnostic Message Service indications received from a single sending device. When the difference exceeds the value of this attribute, a fault in the path from the network interface sending the lower message number is detected
(Unsigned8)

Redundancy flags
This attribute is a bit array that controls how messages are sent and received.
(Unsigned8)

Diagnostic message interval
This attribute defines the time interval in milliseconds between successive sending of the DiagnosticFrames
(Unsigned32)

Aging time
This attribute defines the time interval in milliseconds used by the LRE to remove silent nodes from its Network_Status_Table
(Unsigned32)

DiagnosticFrame send adapter address
This attribute defines the send address of the diagnostic frame.
(OctetString16)

DiagnosticFrame receive adapter address
This attribute defines the receive address of the diagnostic frame.
(OctetString16)

Result(-)

This parameter indicates that the service request failed. The following fields are included in the response.

Invoke ID
Source IP address
Destination IP address
Error info

This parameter specifies the error condition.

7.5.9.2.3 Put redundancy info service

This confirmed service shall be used to retrieve redundancy attributes. After updating the attributes contained in the request, the responder shall return the values and the updated Redundancy Attributes Version number, using the service parameters shown in Table 55.

Table 55 – CRP put redundancy info service

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Invoke ID	M	M(=)		
Source address		M		
Destination address	M	M(=)		
Redundancy attributes version			M	M(=)
Number of network adapters			M	M(=)
Max message number difference			M	M(=)
Redundancy flags			M	M(=)
DiagnosticFrame interval			M	M(=)
Aging time			M	M(=)
DiagnosticFrame adapter send addresses			M	M(=)
DiagnosticFrame adapter receive addresses			M	M(=)
Result (+)			S	S(=)
Invoke ID			M	M(=)
Source address				M
Destination address			M	M(=)
LAN redundancy attributes version			M	M(=)
Number of network interfaces			M	M(=)
Max_sequence_number_difference			M	M(=)
LAN redundancy flags			M	M(=)
DiagnosticFrame interval			M	M(=)
Aging time			M	M(=)
DiagnosticFrame adapter send addresses			M	M(=)
DiagnosticFrame adapter receive addresses			M	M(=)

Parameter name	Req	Ind	Rsp	Cnf
Result (-)			S	S(=)
Invoke ID			M	M(=)
Source address				M
Destination address			M	M(=)
Error info			M	M(=)

Argument

The argument conveys the parameters of the service request:

Result(+)

This selection type parameter indicates that the service request succeeded. The following fields are included in the response.

Invoke ID
Source IP address
Destination IP address
Redundancy attributes version
Number of network adapters
Max message number difference
Redundancy flags
DiagnosticFrame_interval
Aging time
DiagnosticFrame adapter send addresses
DiagnosticFrame adapter receive address

Result(-)

This parameter indicates that the service request failed. The following fields are included in the response.

Invoke ID
Source IP address
Destination IP address

7.5.9.2.4 Get redundancy statistics service

This confirmed service shall be used to retrieve statistics attributes using the service parameters shown in Table 56.

Table 56 – CRP get statistics service

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Invoke ID	M	M(=)		
Source address		M		
Destination address	M	M(=)		
Result (+)			S	S(=)
Invoke ID			M	M(=)
Source address				M
Destination address			M	M(=)
Number of diagnostic message service indications received			M	M(=)
Number of diagnostic message service indications missed			M	M(=)

Parameter name	Req	Ind	Rsp	Cnf
Number of Faults detected			M	M(=)
List of crossed cable status			M	M(=)
Result (-)			S	S(=)
Invoke ID			M	M(=)
Source IP Address				M
Destination IP Address			M	M(=)
Error Info			M	M(=)

Argument

The argument conveys the parameters of the service request:

Invoke ID
Source IP address
Destination IP address

Result(+)

This selection type parameter indicates that the service request succeeded. The following fields are included in the response.

Invoke ID
Source IP address
Destination IP address
Number of diagnostic message service indications received
This attribute counts the number of diagnostics messages received from all devices.
(Unsigned32)

Number of diagnostic message service indications missed
This attribute counts the number of diagnostics message number gaps from all devices. The detection of each missed message number from any device causes this counter to be incremented.
(Unsigned32)

Number of faults detected
This attribute counts the total number of faults detected from missed diagnostics messages from all devices. The detection of a fault from any device causes this counter to be incremented.
(Unsigned32)

List of crossed cable status
This attribute contains the list of crossed cable status. Value 0 means that the list is not present.
(Unsigned32)

Result(-)

This parameter indicates that the service request failed. The following fields are included in the response.

Invoke ID
Source IP address
Destination IP address
Error info

8 BRP – Beacon redundancy protocol

8.1 BRP Overview

This clause specifies a protocol for an Ethernet network tolerant to all single-point failures. This protocol is called beacon redundancy protocol (BRP). A network based on the BRP is called a BRP network. The BRP network is based on switched ISO/IEC 8802-3 (Ethernet) and IEEE 802.1 technologies and redundant infrastructure. In this network, the decision to switch between infrastructures is made individually in each end node.

8.2 BRP Principle of operation

8.2.1 General

The text in 8.2.2 to 8.2.4 is an explanation of the overall actions performed by the BRP state machine. If a difference in the interpretation occurs between this text and the state machines in 8.4, then the state machines take precedence.

8.2.2 Network topology

The BRP network topology can be described as two interconnected top switches, each heading an underlying topology of star, line, or ring. Beacon end nodes shall be connected to the top switches. Examples of star, linear and ring BRP networks are shown in Figure 36, Figure 37 and Figure 38, respectively.

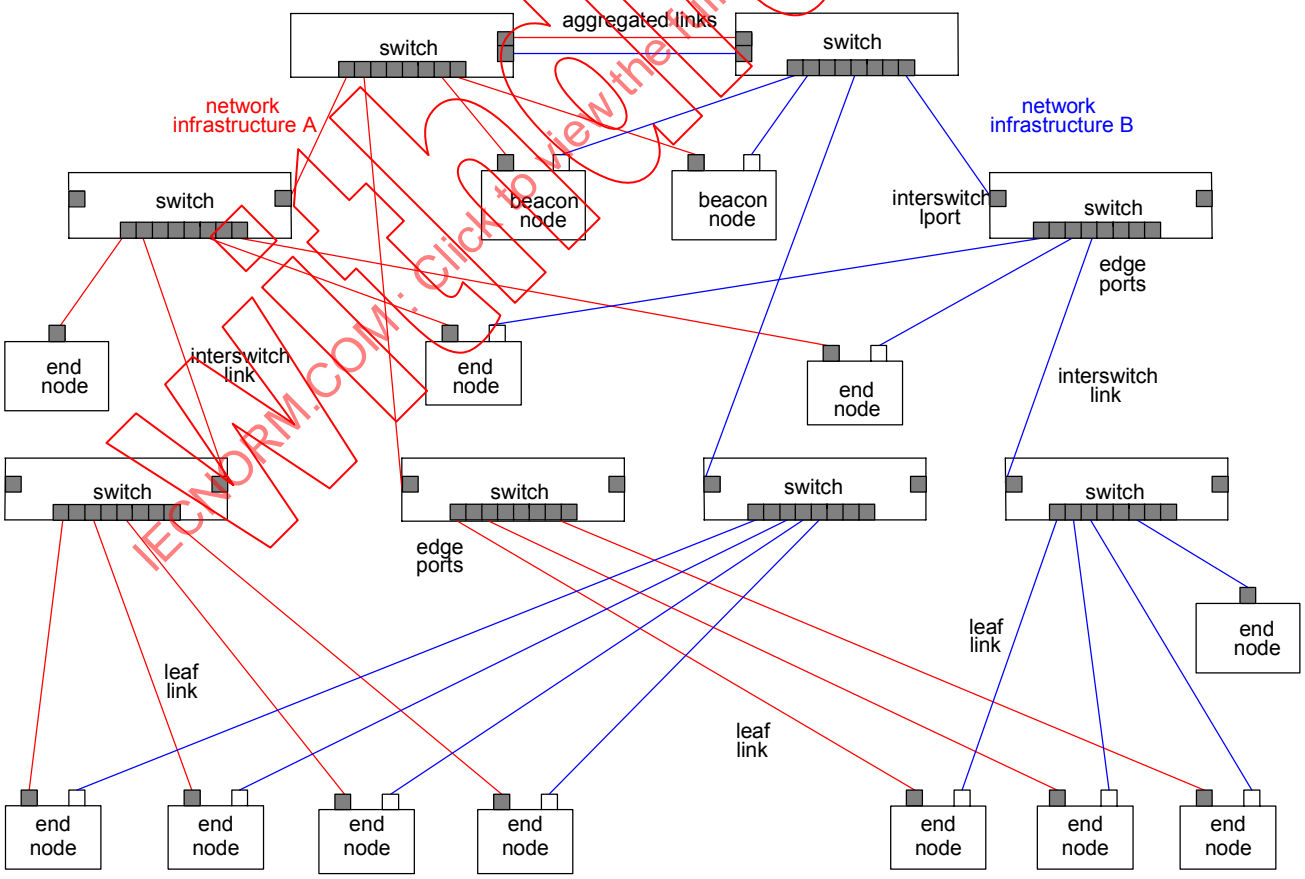


Figure 36 – BRP star network example

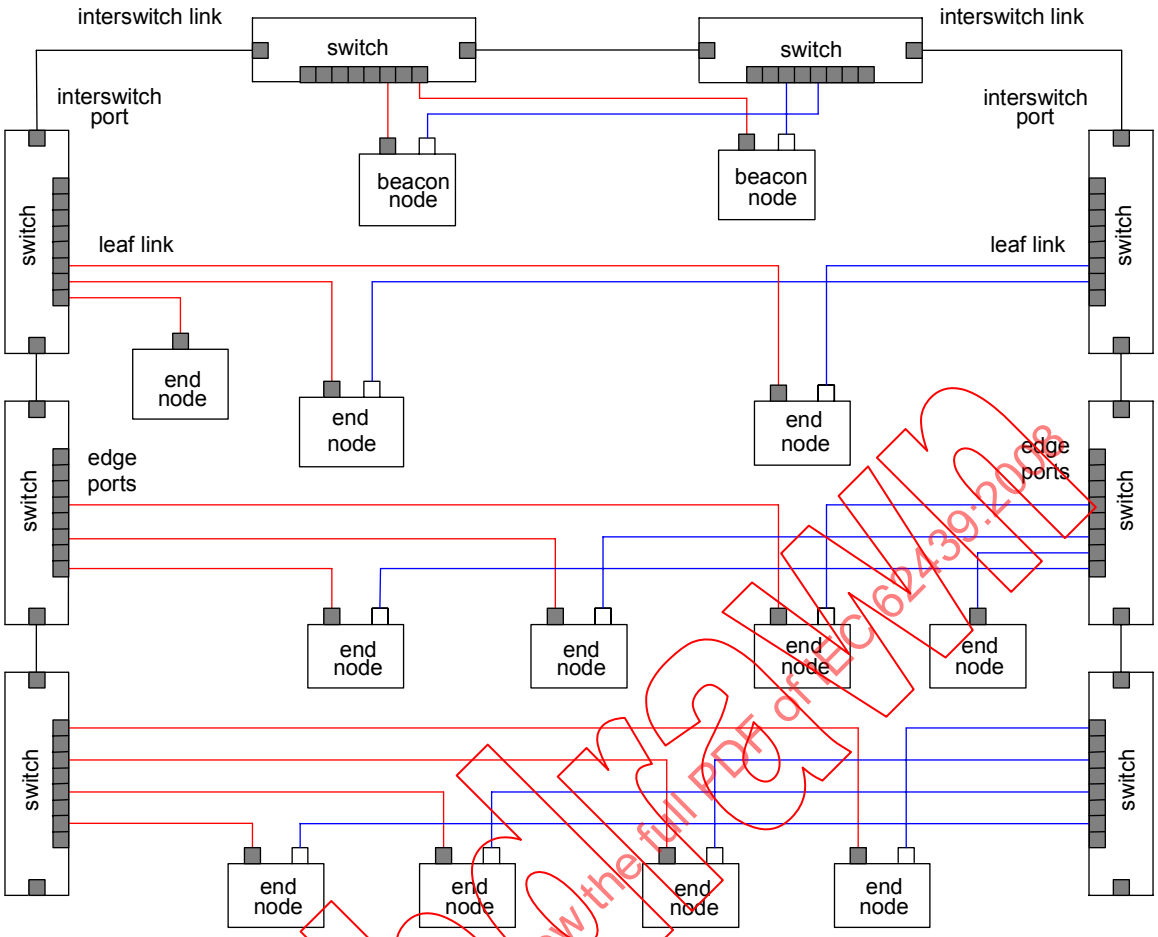


Figure 37 – BRP linear network example

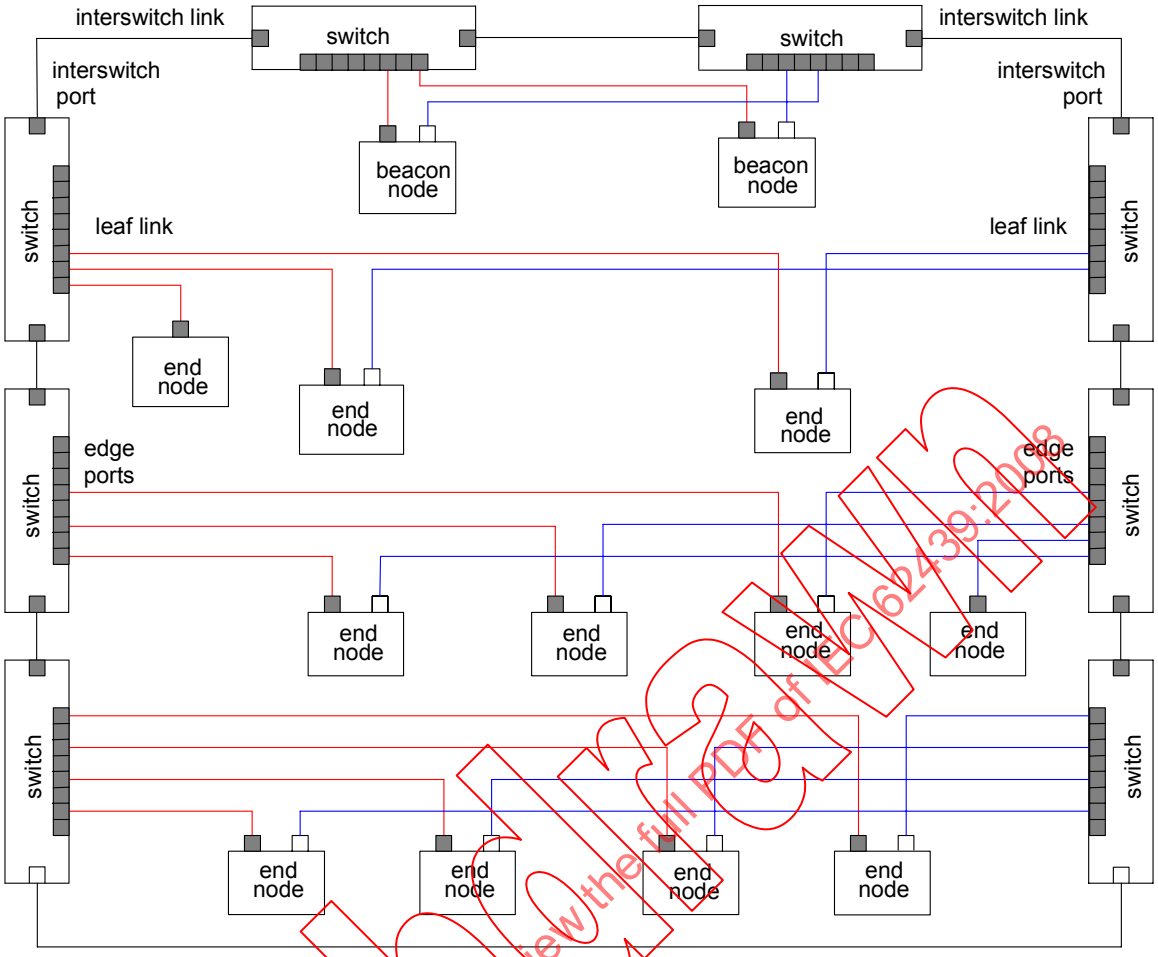


Figure 38 – BRP ring network example

8.2.3 Network components

The BRP network is built from Layer 2 switches compliant with IEEE 802.1D and IEEE 802.3. No support of the BRP protocol in switches is required.

Figure 36 shows an example of a BRP star network in the two-way redundancy mode. It uses two sets of network infrastructure A and B (shown in two different colours). The number of levels of switches and number of switches on each level are dependent only on application requirements. Even with three levels of hierarchy, it is possible to construct very large networks. For example, a BRP star network built from switches with eight regular ports and one uplink port can contain 500 nodes maximum. Two switches at the top level shall be connected to each other with one or more links providing sufficient bandwidth. With link aggregation capability traffic is shared among bundle of links and failure of one link does not bring the network down. With such an arrangement, infrastructures A and B form a single network.

Two types of end nodes can be connected to the BRP network: doubly attached and singly attached. A doubly attached end node can function as a BRP end node or a BRP beacon end node. A BRP beacon end node is a special case of a doubly attached end node that is connected direct to the top switches. Though doubly attached BRP end nodes have two network ports, they use only one MAC address.

At any given point in time, a BRP end node actively communicates through only one of its ports, while blocking all transmit and receive traffic on its other port, with the exception of received Beacon messages and Failure_Notify messages. Fault tolerance is achieved in a

distributed fashion by BRP end nodes switching between their ports from inactive to active mode and vice versa.

As shown in Figure 36, Figure 37 and Figure 38, two beacon end nodes shall be connected to top level switches. Beacon end nodes multi/broadcast a short beacon message on the network periodically. Similarly to BRP end nodes, a beacon end node at any given point in time actively communicates through only one of its ports, while blocking all traffic on its other port, with the exception of received Failure_Notify messages. Fault tolerance is achieved by beacon end nodes switching between their ports from inactive to active mode and vice versa.

Singly attached end nodes may also be connected to the BRP network but they do not support the BRP protocol. A singly attached node can communicate with doubly attached nodes as well as other singly attached nodes on the network.

Since switches are IEEE 802.1D compliant, they support the RSTP protocol. This eliminates loop formation in BRP ring networks like in the one shown in Figure 38.

8.2.4 Rapid reconfiguration of network traffic

For fast reconfiguration, multicast control features in the switches shall be disabled. The multicast traffic is therefore treated as the broadcast traffic.

Unicast packets are affected by switches learning and filtering features. After end node port reconfiguration, switches have invalid knowledge. A switch implementing learning shall update its database when a packet with a learned MAC address in the source field is received on a different port from the learned port stored in the database.

When a BRP end node switches to the inactive port, its first action is to send a short multicast message, called Learning_Update message, through its newly enabled port. As this message propagates through the network, switches update their MAC address database resulting in rapid reconfiguration of the unicast traffic. This message is of no interest to other end nodes in the network and is dropped by them.

8.3 BRP stack and fault detection features

Figure 39 shows the BRP stack architecture. It is applicable to both BRP and beacon end nodes.

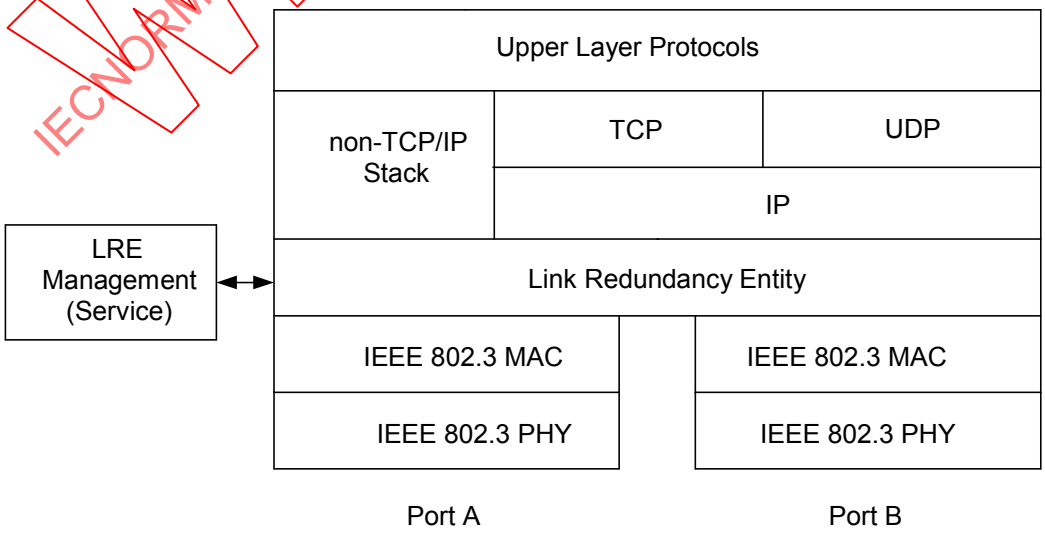


Figure 39 – BRP stack architecture

The BRP stack contains two identical ISO/IEC 8802-3 ports, identified here as ports A and B, connected to the network. These ports interface with the MAC sublayer compliant with ISO/IEC 8802-3. Though there are two physical ports, a BRP end node uses only a single MAC address.

The LRE continuously monitors the status of leaf links between both ports and corresponding ports on the switches. When a failure of the leaf link between the end node active port and the corresponding port on the switch is detected, the LRE shall reconfigure end node ports, provided the inactive port was not in the fault mode as well. After reconfiguration, all traffic flows through the newly activated port. Some messages may be lost during the failure detection and reconfiguration process, and their recovery is supported by upper layer protocols which also deal with messages lost due to other network errors.

The LRE also monitors arrival of beacon messages on both ports. When a beacon message fails to arrive at the active port for a configured timeout period, the port is declared to be in the fault mode, and the LRE shall reconfigure end node ports, provided the other port was not in the fault mode as well. After reconfiguration, all traffic starts flowing through the newly activated port. Failure of beacon messages to arrive at inactive ports shall also be detected.

If one of the top switches fails, then all BRP nodes connected direct to it, or to network infrastructure below it, switch to the other network infrastructure. If, for example, the top switch of the LAN A fails, then all BRP nodes connected to LAN A switch over to LAN B.

If the fault occurred on a beacon end node, the network continues to operate without any problems, since the other beacon end node is active. The rate of the beacon message arrival decreases from approximately two messages per beacon timer interval to one.

It is possible for transmit path failures to occur in the opposite direction to the flow of beacon messages. If such a fault manifests itself in the physical layer, it is detected by end nodes or switches adjacent to the faulty link. This results in a BRP end node reconfiguring its ports immediately or results in traffic being blocked on the affected link. The latter event leads to loss of beacon messages at the downstream end nodes, so they reconfigure themselves at expiry of the beacon timeout.

In a case when such failures are not detectable in the physical layer, the following mechanism is employed by the BRP LRE to detect them. The fault detection method for identifying all transmission failures shall be implemented using lists of communication nodes including a receive timeout value for each transmitting end node of interest to the node. This list may be communicated to the LRE manually or dynamically configured utilizing the LRE management entity.

When a frame from a transmitting end node of interest fails to arrive before expiry of the associated Node_Receive timer, the receiving end node shall send a Failure_Notify message to the transmitting end node and send a Path_Check_Request message to beacon end nodes. Upon reception of a Failure_Notify message, the transmitting end node shall attempt to verify the transmit path by sending the Path_Check_Request message to the beacon end nodes. When the beacon end nodes receive these messages, they shall respond with Path_Check_Response messages. When Path_Check_Request fails to elicit response, an end node shall place its active port in faulted state and activate its inactive port, provided it is not in fault mode as well.

BRP beacon end nodes also behave in a similar way. When a frame from a transmitting end node of interest fails to arrive before expiry of the associated Node_Receive timer, the receiving Beacon end node shall send a Failure_Notify message to the transmitting end node and send a Path_Check_Request message to a designated set of end nodes. When the beacon end nodes receive Failure_Notify messages themselves, they shall verify their transmit path by sending a Path_Check_Request message to a designated set of end nodes. Upon receiving Path_Check_Request message, the designated end nodes shall respond with Path_Check_Response message. When Path_Check_Request fails to elicit response, a

beacon end node shall place its active port in faulted state and activate its inactive port, provided it is not in fault mode as well.

When the faulted port is restored, it shall stay idle until a switchover is initiated or the currently active port fails. When both ports are operational, the BRP end node shall periodically switch its message activity from one port to the other. This switchover is controlled by the Active_Port_Swap timer.

The LRE management entity is used to select an end node type (normal or beacon), configure protocol parameters (for example, beacon timer) and obtain the end node port status (active, failed, idle).

All detected failures shall be reported to the LRE management entity to trigger further diagnosis and repair. Fault diagnostics services shall be provided by LRE management entity or other accessible entities in the network.

8.4 BRP Protocol specification

8.4.1 MAC addresses

BRP protocol shall use multicast address 01-15-4E-00-02-01. Both ports of a BRP node shall have the same MAC address for active communication.

8.4.2 EtherType

The BRP protocol shall use assigned EtherType 0x80E1.

8.4.3 Fault detection mechanisms

The following fault detection mechanisms are used.

- Link fault detection

  This mechanism covers physical layer failures in transmit and receives directions on a link connected direct to the end node.

- Receive path fault detection

  This is accomplished utilizing the beacon message transmission mechanism.

- Transmit path fault detection

  This is accomplished utilizing Failure_Notify, Path_Check_Request, and Path_Check_Response messages. The periodic switchover between active and inactive ports ensures coverage of all transmit paths in the network.

8.4.4 End node state diagram

The BRP end node state diagram is shown in Figure 40.

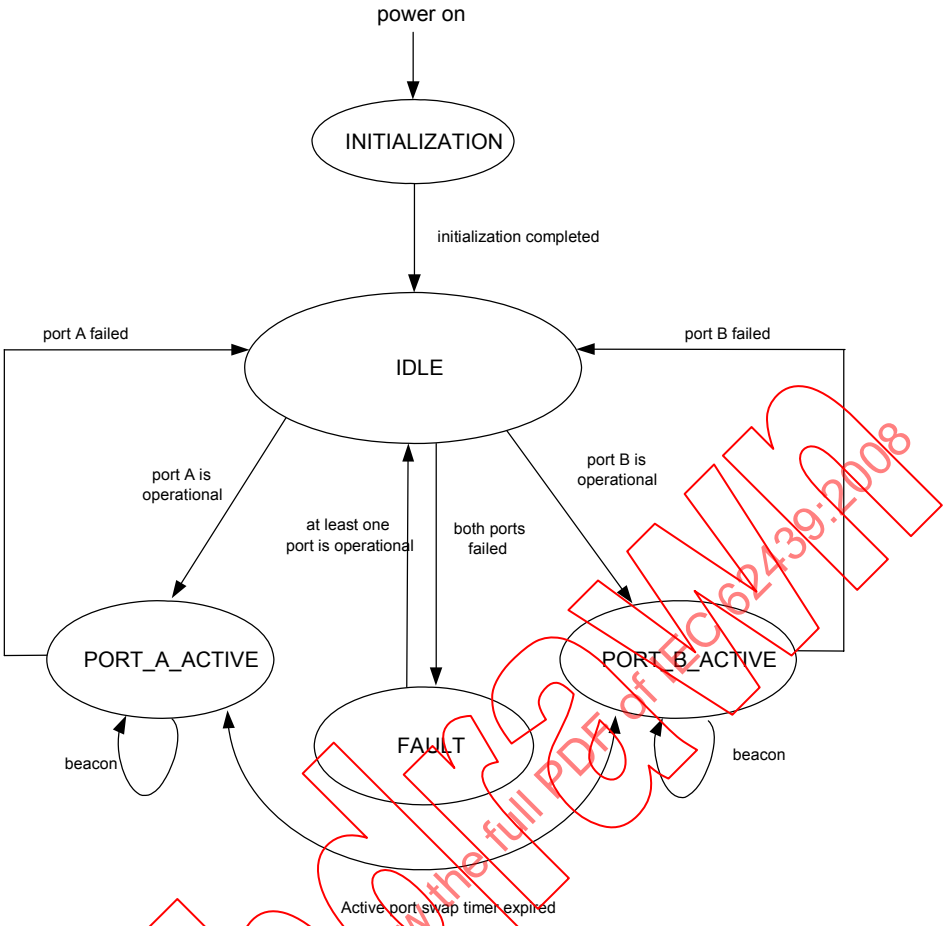


Figure 40 – BRP state diagram of end node

The BRP end node protocol state machine shall perform in accordance with the state transition table presented in Table 58.

When the node is powered up and has passed the initialization process (the Initialization_Completed flag is set), it resets the protocol state machine and transitions to the IDLE state.

Since Port_A_Failed and Port_B_Failed flags were initially set, the node immediately transitions from the IDLE to the FAULT state.

If link A is active and a beacon message is received on this link, then the node transitions from the FAULT state back to the IDLE. A Learning_Update message is generated on this port and the node transitions from the IDLE to the PORT_A_ACTIVE state.

The node tests port B simultaneously with port A using the procedure described above. If both ports are operational, either one can be selected as the default.

Periodic reception of beacon messages (Beacon_A_Received is set) keeps the node in the PORT_A_ACTIVE state and trigger reset of the No_Beacon_A timer.

If, when in the PORT_A_ACTIVE state, link A becomes inactive (Link_A_Active is reset) or no beacon messages were received for a given time period (No_Beacon_A timer expired and Beacon_A_Received is reset), the node sets the Port_A_Failed flag and transitions to the IDLE state where it attempts to switch to port B.

Operation of port B is identical to operation of port A.

If a Node_Receive timer expires, the receiving node sends a Failure_Notify message to the associated transmitting end node and sends Path_Check_Request message on its active port to beacon end nodes. When the transmitting end node receives the Failure_Notify message it attempts to verify transmission path on its active port by sending a Path_Check_Request message on this port to beacon end nodes. When beacon nodes receive these messages, they issue Path_Check_Response messages addressed to the requesting node.

When Path_Check_Request fails to elicit response (Path_A_Check/Path_B_Check timer expired), the node sets the Path_A_Failed/Path_B_Failed flag and Port_A_Failed/ Port_B_Failed flag, and transitions to the IDLE state where it attempts to switch to port B/A.

If both ports failed, then the node transitions from the IDLE to the FAULT state and stays there until one of the ports becomes operational. In FAULT state, a node continuously monitors link status (Link_A_Active/Link_B_Active flags) and beacon arrival status (Beacon_A_Received/Beacon_B_Received flags). If Path_A_Failed and/or Path_B_Failed flags were set, the node also sends Path_Check_Request and monitors arrival of Path_Check_Response message for corresponding ports. When one of the ports becomes operational (Port_A_Failed/Port_B_Failed is reset), the node transitions back to the IDLE state and then to PORT_A_ACTIVE/PORT_B_ACTIVE as appropriate.

When a node receives a Path_Check_Request message in PORT_A_ACTIVE or PORT_B_ACTIVE states, it responds with the Path_Check_Response message and stays in current state.

When in PORT_A_ACTIVE/PORT_B_ACTIVE state and the Active_Port_Swap timer expires, the node transitions to PORT_B_ACTIVE/PORT_A_ACTIVE state provided PORT_B_FAILED/PORT_A_FAILED is not set.

The No_Beacon timer period is a configuration parameter selected for a specific system. The mandatory default value of the beacon period is 450 µs resulting in the default value of the No_Beacon period of 950 µs. The timeout period is chosen such that at least two beacon messages from each beacon end node have to be lost before fault is declared on a port.

A BRP compliant end node shall be able to receive beacon messages over both of its ports sent from both beacon end nodes at the mandatory default value of the beacon period.

The Path_A_Check and Path_B_Check timer periods are configuration parameters selected for a specific system. The mandatory default value is 2 ms.

The Active_Port_Swap timer period is a configuration parameter selected for a specific system. The mandatory default value is 1 h.

Table 57 specifies the flags used in the BRP end node state machine.

Table 57 – BRP end node flags

Name	Description	Data Type
Initialization_Completed	Used to indicate initialization completed successfully	**BOOL**
Link_A_Active	Used to indicate physical layer link status of Port A	**BOOL**
Beacon_A_Received	Used to indicate beacon message was received on Port A	**BOOL**
Path_A_Failed	Used to indicate if Path_Check_Response message was received for Path_Check_Request message on Port A	**BOOL**
Link_B_Active	Used to indicate physical layer link status of Port B	**BOOL**
Beacon_B_Received	Used to indicate beacon message was received on Port B	**BOOL**
Path_B_Failed	Used to indicate if Path_Check_Response message was received for Path_Check_Request message on Port B	**BOOL**

Name	Description	Data Type
Path_A_Request	Used to indicate if Path_Check_Request message was sent on Port A	**BOOL**
Path_B_Request	Used to indicate if Path_Check_Request message was sent on Port B	**BOOL**
Port_A_Failed	Used to indicate if Port A has failed	**BOOL**
Port_B_Failed	Used to indicate if Port B has failed	**BOOL**

Table 58 specifies the BRP end node state transition table.

Table 58 – BRP end node state transition table

#	Current State	Event /Condition =>Action	Next State
1	INITIALIZATION	**Initialization is completed** => Set Initialization_Completed Reset Link_A_Active Reset Beacon_A_Received Stop No_Beacon_A timer Reset Path_A_Failed Stop Path_A_Check timer Reset Path_A_Request Reset Link_B_Active Reset Beacon_B_Received Stop No_Beacon_B timer Reset Path_B_Failed Stop Path_B_Check timer Reset Path_B_Request Set Port_A_Failed Set Port_B_Failed Stop Node_Receive timers Stop Active_Port_Swap timer	IDLE
2	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Port A Link Pass Status** => Set Link_A_Active	STAY IN CURRENT STATE
3	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Port A Link Fail Status** => Reset Link_A_Active	STAY IN CURRENT STATE
4	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Port B Link Pass Status** => Set Link_B_Active	STAY IN CURRENT STATE
5	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Port B Link Fail Status** => Reset Link_B_Active	STAY IN CURRENT STATE

#	Current State	Event /Condition =>Action	Next State
6	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Beacon Message Received on Port A** => Set Beacon_A_Received Start No_Beacon_A timer	STAY IN CURRENT STATE
7	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**No_Beacon_A timer expired** => Reset Beacon_A_Received	STAY IN CURRENT STATE
8	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Beacon Message Received on Port B** => Set Beacon_B_Received Start No_Beacon_B timer	STAY IN CURRENT STATE
9	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**No_Beacon_B timer expired** => Reset Beacon_B_Received	STAY IN CURRENT STATE
10	PORT_A_ACTIVE	**Failure_Notify message is received** => Send Path_Check_Request message on Port A Set Path_A_Request Start Path_A_Check timer	PORT_A_ACTIVE
11	PORT_B_ACTIVE	**Failure_Notify message is received** => Send Path_Check_Request message on Port B Set Path_B_Request Start Path_B_Check timer	PORT_B_ACTIVE
12	PORT_A_ACTIVE, FAULT	**Path_A_Check timer expired** => Set Path_A_Failed Reset Path_A_Request	STAY IN CURRENT STATE
13	PORT_B_ACTIVE, FAULT	**Path_B_Check timer expired** => Set Path_B_Failed Reset Path_A_Request	STAY IN CURRENT STATE
14	PORT_A_ACTIVE, FAULT	**Path_Check_Response message is received on Port A** => Stop Path_A_Check timer Reset Path_A_Failed Reset Path_A_Request	STAY IN CURRENT STATE
15	PORT_B_ACTIVE, FAULT	**Path_Check_Response message is received on Port B** => Stop Path_B_Check timer Reset Path_B_Failed Reset Path_B_Request	STAY IN CURRENT STATE

#	Current State	Event /Condition =>Action	Next State
16	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Link_A_Active is set** **AND** **Beacon_A_Received is set** **AND** **Path_A_Failed is reset** => Reset Port_A_Failed	STAY IN CURRENT STATE
17	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Link_A_Active is reset** **OR** **Beacon_A_Received is reset** **OR** **Path_A_Failed is set** => Set Port_A_Failed	STAY IN CURRENT STATE
18	IDLE	**Port_A_Failed is Reset** => Send Learning_Update message on Port A Start Node_Receive timers Start Active_Port_Swap timer	PORT_A_ACTIVE
19	PORT_A_ACTIVE	**Port_A_Failed is Set** => Stop Path_A_Check timer Reset Path_A_Request Stop Node_Receive timers Stop Active_Port_Swap timer	IDLE
20	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Link_B_Active is set** **AND** **Beacon_B_Received is set** **AND** **Path_B_Failed is reset** => Reset Port_B_Failed	STAY IN CURRENT STATE
21	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Link_B_Active is reset** **OR** **Beacon_B_Received is reset** **OR** **Path_B_Failed is set** => Set Port_B_Failed	STAY IN CURRENT STATE
22	IDLE	**Port_B_Failed is Reset** => Send Learning_Update message on Port B Start Node_Receive timers Start Active_Port_Swap timer	PORT_B_ACTIVE
23	PORT_B_ACTIVE	**Port_B_Failed is Set** => Stop Path_B_Check timer Reset Path_B_Request Stop Node_Receive timers Stop Active_Port_Swap timer	IDLE

#	Current State	Event /Condition =>Action	Next State
24	IDLE	**Port_A_Failed is Set** **AND** **Port_B_Failed is set**	FAULT
25	FAULT	**Link_A_Active is set** **AND** **Beacon_A_Received is set** **AND** **Path_A_Failed is set** **AND** **Path_A_Request is reset** => Set Path_A_Request Send Path_Check_Request message on Port A Start Path_A_Check timer	FAULT
26	FAULT	**Link_B_Active is set** **AND** **Beacon_B_Received is set** **AND** **Path_B_Failed is set** **AND** **Path_B_Request is reset** => Set Path_B_Request Send Path_Check_Request message on Port B Start Path_B_Check timer	FAULT
27	FAULT	**Port_A_Failed is reset** **OR** **Port_B_Failed is reset**	IDLE
28	PORT_A_ACTIVE, PORT_B_ACTIVE	**Path_Check_Request message received on active port** => Send Path_Check_Response message on active port	STAY IN CURRENT STATE
29	PORT_A_ACTIVE, PORT_B_ACTIVE	**Frame received from transmit node of interest on active port** => Restart associated Node_Receive timer	STAY IN CURRENT STATE
30	PORT_A_ACTIVE	**Node_Receive timer expired** => Send Failure_Notify message on Port A to associated transmit node Send Path_Check_Request message on Port A Set Path_A_Request Start Path_A_Check timer	PORT_A_ACTIVE
31	PORT_B_ACTIVE	**Node_Receive timer expired** => Send Failure_Notify message on Port B to associated transmit node Send Path_Check_Request message on Port B Set Path_B_Request Start Path_B_Check timer	PORT_B_ACTIVE

#	Current State	Event /Condition =>Action	Next State
32	PORT_A_ACTIVE	**Active_Port_Swap timer expired** **AND** **Port_B_Failed is reset** => Stop Path_A_Check timer Reset Path_A_Request Send Learning_Update message on Port B Start Active_Port_Swap timer	PORT_B_ACTIVE
33	PORT_A_ACTIVE	**Active_Port_Swap timer expired** **AND** **Port_B_Failed is set** => Start Active_Port_Swap timer	PORT_A_ACTIVE
34	PORT_B_ACTIVE	**Active_Port_Swap timer expired** **AND** **Port_A_Failed is reset** => Stop Path_B_Check timer Reset Path_B_Request Send Learning_Update message on Port A Start Active_Port_Swap timer	PORT_A_ACTIVE
35	PORT_B_ACTIVE	**Active_Port_Swap timer expired** **AND** **Port_A_Failed is set** => Start Active_Port_Swap timer	PORT_B_ACTIVE

8.4.5 Beacon end node state diagram

If the end node is configured as a beacon, it periodically generates Beacon messages. The beacon end node state diagram is shown in Figure 41.

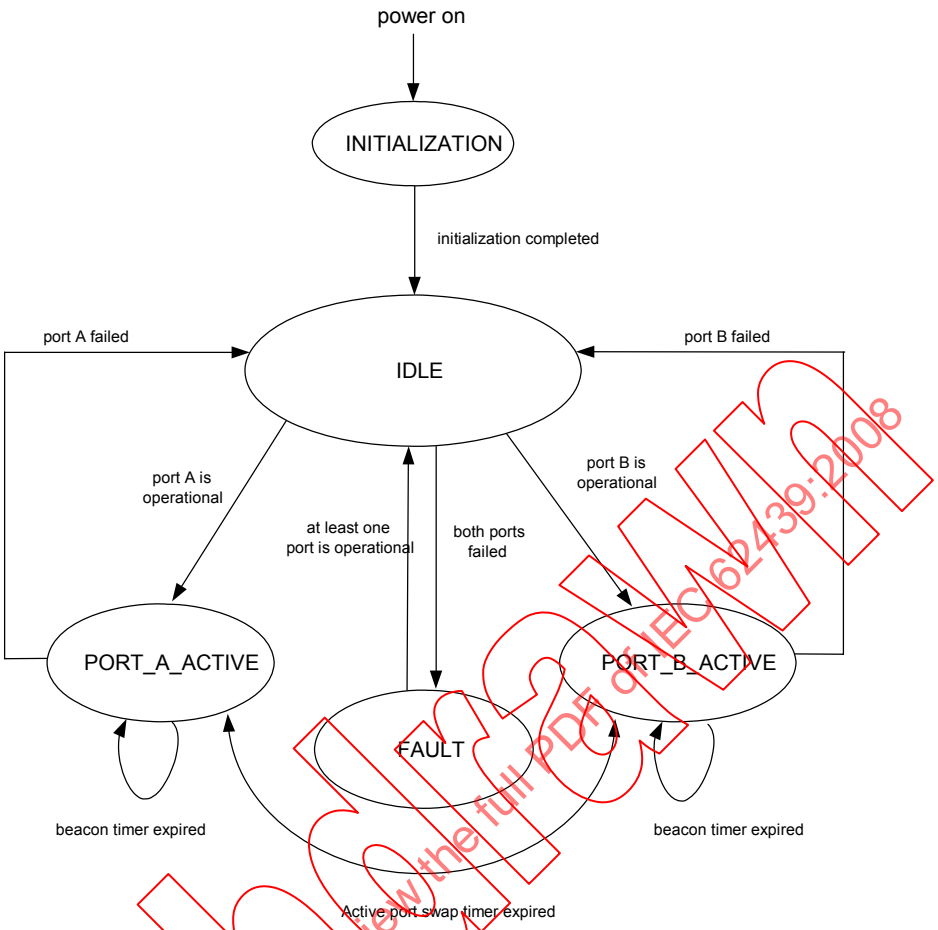


Figure 41 – BRP state diagram for beacon end node

When the beacon end node is powered up and has passed the initialization process (the Initialization_Completed flag is set), it resets the protocol state machine and transitions to the IDLE state.

Since Port_A_Failed and Port_B_Failed flags were initially set, the node immediately transitions from the IDLE to the FAULT state.

If link A is active, then the node transitions from the FAULT state back to the IDLE. The node then generates a beacon message on port A, starts the beacon timer and transitions to the PORT_A_ACTIVE state.

The node tests port B simultaneously with port A executing procedure identical to the one described above. If both ports are operational, either one can be selected by default.

When the beacon timer expires the node transmits the beacon message, restarts the beacon timer and continues staying in the PORT_A_ACTIVE state.

If, when in the PORT_A_ACTIVE state, link A becomes inactive (Link_A_Active is reset), the node sets the Port_A_Failed flag, stops the beacon timer and transitions to the IDLE state where it attempts to switch to port B.

Operation of port B is identical to operation of port A.

If a Node_Receive_Timer expires, the receiving beacon end node sends a Failure_Notify message to the associated transmitting end node and sends Path_Check_Request message on its active port to designated set of end nodes. When the transmitting end node receives the Failure_Notify message it attempts to verify transmission path as described in 8.4.4.

When a beacon end node receives Failure_Notify message it attempts to verify transmission path on its active port by sending Path_Check_Request message on this port to designated set of nodes. When the designated set of nodes receives this message, they respond with Path_Check_Response message.

When Path_Check_Request fails to elicit response (Path_A_Check/Path_B_Check timer expired), the node sets the Path_A_Failed/Path_B_Failed flag and Port_A_Failed/ Port_B_Failed flag and transitions to the IDLE state where it attempts to switch to port B/A.

If both ports failed, then the node transitions from the IDLE to the FAULT state and stays there until one of the ports becomes operational. In FAULT state, a node continuously monitors link status (Link_A_Active/Link_B_Active flags). If Path_A_Failed and/or Path_B_Failed flags were set, the node also sends Path_Check_Request and monitors arrival of Path_Check_Response message for corresponding ports. When one of the ports becomes operational (Port_A_Failed/Port_B_Failed is reset) the node transitions back to the IDLE state and then to PORT_A_ACTIVE/PORT_B_ACTIVE as appropriate.

When a node receives a Path_Check_Request message in PORT_A_ACTIVE or PORT_B_ACTIVE states, it responds with the Path_Check_Response message and stays in current state.

When in PORT_A_ACTIVE/PORT_B_ACTIVE state and the Active_Port_Swap timer expires, the node transitions to PORT_B_ACTIVE/PORT_A_ACTIVE state provided PORT_B_FAILED/PORT_A_FAILED is not set.

The No_Beacon timer period is a configuration parameter selected for a specific system. The mandatory default value of the beacon period is 450 µs resulting in the default value of the No_Beacon period of 950 µs. The timeout period is chosen such that at least two beacon messages from each beacon end node have to be lost before fault is declared on a port.

A BRP compliant beacon end node shall be able to broadcast the beacon message every 450 microseconds via its active port.

The Path_A_Check and Path_B_Check timer periods are configuration parameters selected for a specific system. The mandatory default value is 2 ms.

The Active_Port_Swap timer period is a configuration parameter selected for a specific system. The mandatory default value is 1 h.

Table 59 specifies the flags used in the BRP beacon end node state machine.

Table 59 – BRP beacon end node flags

Name	Description	Data Type
Initialization_Completed	Used to indicate initialization completed successfully	BOOL
Link_A_Active	Used to indicate Physical layer link status of Port A	BOOL
Path_A_Failed	Used to indicate if Path_Check_Response message was received for Path_Check_Request message on Port A	BOOL
Link_B_Active	Used to indicate Physical layer link status of Port B	BOOL
Path_B_Failed	Used to indicate if Path_Check_Response message was received for Path_Check_Request message on Port B	BOOL

Name	Description	Data Type
Path_A_Request	Used to indicate if Path_Check_Request message was sent on Port A	BOOL
Path_B_Request	Used to indicate if Path_Check_Request message was sent on Port B	BOOL
Port_A_Failed	Used to indicate if Port A has failed	BOOL
Port_B_Failed	Used to indicate if Port B has failed	BOOL

Table 60 specifies the BRP Beacon end node state transition table.

Table 60 – BRP beacon end node state transition table

#	Current state	Event /Condition =>Action	Next state
1	INITIALIZATION	**Initialization is completed** => Set Initialization_Completed Reset Link_A_Active Reset Path_A_Failed Stop Path_A_Check timer Reset Path_A_Request Reset Link_B_Active Reset Path_B_Failed Stop Path_B_Check timer Reset Path_B_Request Set Port_A_Failed Set Port_B_Failed Stop Node_Receive timers Stop Active_Port_Swap timer	IDLE
2	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Port A Link Pass Status** => Set Link_A_Active	STAY IN CURRENT STATE
3	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Port A Link Fail Status** => Reset Link_A_Active	STAY IN CURRENT STATE
4	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Port B Link Pass Status** => Set Link_B_Active	STAY IN CURRENT STATE
5	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Port B Link Fail Status** => Reset Link_B_Active	STAY IN CURRENT STATE
6	PORT_A_ACTIVE, PORT_B_ACTIVE	**Frame received from transmit node of interest on active port** => Restart associated Node_Receive timer	STAY IN CURRENT STATE

#	Current state	Event /Condition =>Action	Next state
7	PORT_A_ACTIVE	**Node_Receive timer expired** => Send Failure_Notify message on Port A to associated transmit node Send Path_Check_Request message on Port A Set Path_A_Request Start Path_A_Check timer	PORT_A_ACTIVE
8	PORT_B_ACTIVE	**Node_Receive timer expired** => Send Failure_Notify message on Port B to associated transmit node Send Path_Check_Request message on Port B Set Path_B_Request Start Path_B_Check timer	PORT_B_ACTIVE
9	PORT_A_ACTIVE	**Failure_Notify message is received** => Send Path_Check_Request message on Port A Set Path_A_Request Start Path_A_Check timer	PORT_A_ACTIVE
10	PORT_B_ACTIVE	**Failure_Notify message is received** => Send Path_Check_Request message on Port B Set Path_B_Request Start Path_B_Check timer	PORT_B_ACTIVE
11	PORT_A_ACTIVE, FAULT	**Path_A_Check timer expired** => Set Path_A_Failed Reset Path_A_Request	STAY IN CURRENT STATE
12	PORT_B_ACTIVE, FAULT	**Path_B_Check timer expired** => Set Path_B_Failed Reset Path_A_Request	STAY IN CURRENT STATE
13	PORT_A_ACTIVE, FAULT	**Path_Check_Response message is received on Port A** => Stop Path_A_Check timer Reset Path_A_Failed Reset Path_A_Request	STAY IN CURRENT STATE
14	PORT_B_ACTIVE, FAULT	**Path_Check_Response message is received on Port B** => Stop Path_B_Check timer Reset Path_B_Failed Reset Path_B_Request	STAY IN CURRENT STATE

#	Current state	Event /Condition =>Action	Next state
15	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Link_A_Active is set** **AND** **Path_A_Failed is reset** => Reset Port_A_Failed	STAY IN CURRENT STATE
16	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Link_A_Active is reset** **OR** **Path_A_Failed is set** => Set Port_A_Failed	STAY IN CURRENT STATE
17	IDLE	**Port_A_Failed is Reset** => Send Beacon message on Port A Start Beacon timer Start Node_Receive timers Start Active_Port_Swap timer	PORT_A_ACTIVE
18	PORT_A_ACTIVE	**Port_A_Failed is Set** => Stop Beacon timer Stop Path_A_Check timer Reset Path_A_Request Stop Node_Receive timers Stop Active_Port_Swap timer	IDLE
19	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Link_B_Active is set** **AND** **Path_B_Failed is reset** => Reset Port_B_Failed	STAY IN CURRENT STATE
20	IDLE, FAULT, PORT_A_ACTIVE, PORT_B_ACTIVE	**Link_B_Active is reset** **OR** **Path_B_Failed is set** => Set Port_B_Failed	STAY IN CURRENT STATE
21	IDLE	**Port_B_Failed is Reset** => Send Beacon message on Port B Start Beacon timer Start Node_Receive timers Start Active_Port_Swap timer	PORT_B_ACTIVE
22	PORT_B_ACTIVE	**Port_B_Failed is Set** => Stop Beacon timer Stop Path_B_Check timer Reset Path_B_Request Stop Node_Receive timers Stop Active_Port_Swap timer	IDLE

#	Current state	Event /Condition =>Action	Next state
23	IDLE	**Port_A_Failed is Set** **AND** **Port_B_Failed is set**	FAULT
24	FAULT	**Link_A_Active is set** **AND** **Path_A_Failed is set** **AND** **Path_A_Request is reset** => Set Path_A_Request Send Path_Check_Request message on Port A Start Path_A_Check timer	FAULT
25	FAULT	**Link_B_Active is set** **AND** **Path_B_Failed is set** **AND** **Path_B_Request is reset** => Set Path_B_Request Send Path_Check_Request message on Port B Start Path_B_Check timer	FAULT
26	FAULT	**Port_A_Failed is reset** **OR** **Port_B_Failed is reset**	IDLE
27	PORT_A_ACTIVE, PORT_B_ACTIVE	**Path_Check_Request message received on active port** => Send Path_Check_Response message on active port	STAY IN CURRENT STATE
28	PORT_A_ACTIVE, PORT_B_ACTIVE	**Beacon timer expired** => Transmit beacon message on active port Start beacon timer	STAY IN CURRENT STATE
29	PORT_A_ACTIVE	**Active_Port_Swap timer expired** **AND** **Port_B_Failed is reset** **=>** Stop Path_A_Check timer Reset Path_A_Request Start Active_Port_Swap timer	PORT_B_ACTIVE
30	PORT_A_ACTIVE	**Active_Port_Swap timer expired** **AND** **Port_B_Failed is set** **=>** Start Active_Port_Swap timer	PORT_A_ACTIVE

#	Current state	Event /Condition =>Action	Next state
31	PORT_B_ACTIVE	**Active_Port_Swap timer expired** **AND** **Port_A_Failed is reset** **=>** Stop Path_B_Check timer Reset Path_B_Request Start Active_Port_Swap timer	PORT_A_ACTIVE
32	PORT_B_ACTIVE	**Active_Port_Swap timer expired** **AND** **Port_A_Failed is set** **=>** Start Active_Port_Swap timer	PORT_B_ACTIVE

8.5 BRP Message structure

8.5.1 General

The BRP messages contain header, payload and the IEEE 802.3 FCS.

In Table 61 to Table 66:

- the destination MAC Address is the multicast address defined in 8.4.1 for beacon and Learning_Update messages, while Failure_Notify, Path_Check_Request and Path_Check_Response messages use unicast addresses of receivers;
- IEEE 802.1Q tag priority = 7 (highest priority);
- all multi-byte fields shall be encoded in big endian (except the Ethernet addresses).

8.5.2 IEEE 802.3 tagged frame header

Table 61 specifies the format for the common header with IEEE 802.3 tagged frame.

NOTE The tag frame priority should be preserved when the BRP message is transferred through the LAN.

Table 61 – BRP common header with IEEE 802.3 tagged frame format

Byte	Field	Type	Remarks
0	Destination MAC Address	UINT8[6]	
6	Source MAC Address	UINT8[6]	
12	802.1Q Tag Type	UINT16	= 0x8100
14	802.1Q Tag control	UINT16	= 0xE000 + optional VLAN_ID
16	BRP EtherType	UINT16	= 0x80E1
18	BRP Sub-type	UINT8	= 0x01
19	BRP Version	UINT8	= 0x01

8.5.3 Beacon message

Table 62 specifies the format for the beacon message.

Table 62 – BRP beacon message format

Byte	Field	Type	Remarks
20	Message type	UINT8	= 0x80
21	Source IP address	UINT32	= 0x0, if source has no IP address
25	Sequence Id	UINT32	
29	Beacon timeout	UINT32	In µs
33	Reserved	UINT8[31]	
64	CRC	UINT32	

8.5.4 Learning_Update message

Table 63 specifies the format for the Learning_Update message.

Table 63 – BRP Learning_Update message format

Byte	Field	Type	Remarks
20	Message type	UINT8	= 0x40
21	Source IP address	UINT32	= 0x0, if source has no IP address
25	Sequence Id	UINT32	
29	Reserved	UINT8[35]	
64	CRC	UINT32	

8.5.5 Failure_Notify message

Table 64 specifies the format for the Failure_Notify message.

Table 64 – BRP Failure_Notify message format

Byte	Field	Type	Remarks
20	Message type	UINT8	= 0x20
21	Source IP address	UINT32	= 0x0, if source has no IP address
25	Sequence Id	UINT32	
29	Reserved	UINT8[35]	
64	CRC	UINT32	

8.5.6 Path_Check_Request message

Table 65 specifies the format for the Path_Check_Request message.

Table 65 – BRP Path_Check_Request message format

Byte	Field	Type	Remarks
20	Message type	UINT8	= 0x10
21	Source IP address	UINT32	= 0x0, if source has no IP address
25	Sequence Id	UINT32	
29	Source Port	UINT8	= 0x1 or 0x2
30	Reserved	UINT8[34]	
64	CRC	UINT32	

8.5.7 Path_Check_Response message

Table 66 specifies the format for the Path_Check_Response message.

Table 66 – BRP Path_Check_Response message format

Byte	Field	Type	Remarks
20	Message type	UINT8	= 0x08
21	Source IP address	UINT32	= 0x0, if source has no IP address
25	Sequence Id	UINT32	= Sequence Id of Path_Check_Request message
29	Source port	UINT8	= Source Port of Path_Check_Request message
30	Reserved	UINT8[34]	
64	CRC	UINT32	

8.6 BRP Fault recovery time

The following types of faults may occur in an BRP-based network.

- Leaf link faults. These faults are detectable in the end node physical layer. The fault recovery time shall be less than 10 µs.
- Faults occurred in the direction of flow of beacon messages plus those occurred in the opposite direction to the flow of beacon messages but are detectable in the node/switch physical layer. The fault recovery time in this case is two beacon timeouts which is less than 1 ms.
- Faults occurred in the opposite direction to the flow of beacon messages but are not detectable in the node/switch physical layer. Since the fault recovery time in this case is longer than in the two cases described above, it is considered the worst case.

NOTE Faults in the inactive paths transmitting towards the beacon have no effect on operational performance until the next network switchover. At switchover, they are detected using the above methods with the given worst-case recovery time.

The worst-case fault recovery time is:

$t_{fr} = t_{nr} + t_{id} + t_{pcr}$,

where

t_{fr} is the fault recovery time;
t_{nr} is the Node_Receive timer time out;
t_{id} is the infrastructure propagation delay of the Failure_Notify message;
t_{pcr} is the path check request timer time out.

EXAMPLE

Consider a network of 500 nodes with 3 layers of 8-port switches, similar to the one shown in Figure 36.

Assuming that all links have a data rate of 100 Mbit/s and a data frame size of 1 522 octets, the data frame transmit time plus inter-frame gap time is about 124 µs.

The Failure_Notify message size is 68 octets, its transmit time plus inter-frame gap time is about 8 µs.

Assuming the worst-case message queuing in the switch, the Failure_Notify message delay in each switch is:

124 µs + 8 µs = 132 µs.

The total delay of the Failure_Notify message travelling through the longest path of the network infrastructure is:

$$t_{id} = 8\ \mu s + (132 \times 6)\ \mu s + 8\ \mu s = 808\ \mu s = 0{,}81\ ms.$$

Assuming that Node_Receive timer time out t_{nr} = 2 ms, and Path_Check_A_Request timer time out t_{pcr} = 2 ms, and also assuming that Path_Check_B_Request timer is set to the same time as the Path_Check_A_Request timer:

$$t_{fr} = 2\ ms + 0{,}81\ ms + 2\ ms = 4{,}81\ ms.$$

8.7 BRP Service definition

8.7.1 Supported services

The BRP services provide ability to set end node parameters and read these parameters and node status. The following services are provided:

- set node parameters;
- get node parameters;
- add node receive parameters;
- remove node receive parameters;
- get node status.

8.7.2 Common service parameters

The following service parameters are common to several BRP services.

Node Name
This parameter contains the end node name.

(String32)

Source MAC address
This parameter is the MAC address of the node from which the service request has been sent.

(String16)

Destination MAC address
This parameter is the MAC address of the node to which the service request has been sent.

(String16)

Node type
This parameter contains description of the end node type (DANB or beacon).

(String32)

VLAN ID
This parameter contains the VLAN identifier.

(String32)

Status
This parameter contains description of the positive response to a service request.

(String128)

Error info
This parameter contains description of the negative response to a service request.

(String128)

8.7.3 Set node parameters service

Table 67 shows the parameters of the set node parameters service.

Table 67 – BRP set node parameters service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Node name	M	M(=)		
Source MAC address	M	M(=)		
Destination MAC address	M	M(=)		
Node type	M	M(=)		
Beacon timer reload value	C	C(=)		
No_Beacon timer reload value	C	C(=)		
Path_A_Check timer reload value	M	M(=)		
Path_B_Check timer reload value	M	M(=)		
Active_Port_Swap timer reload value	M	M(=)		
Number of designated nodes	C	C(=)		
Designated node list	C	C(=)		
VLAN ID	C	C(=)		
Result (+)			S	S(=)
Node name			M	M(=)
Source MAC address			M	M(=)
Destination MAC address			M	M(=)
Status			M	M(=)
Result (-)			S	S(=)
Node name			M	M(=)
Source MAC address			M	M(=)
Destination MAC address			M	M(=)
Error info			M	M(=)

Argument
The argument conveys the parameters of the service request.

Beacon timer reload value
This parameter contains the value of the beacon timer in microseconds.

(Unsigned 32)

No_Beacon timer reload value
This parameter contains the value of the No_Beacon timer in microseconds.

(Unsigned 32)

Path_A_Check timer reload value
This parameter contains the value of the Path_A_Check timer in microseconds.

(Unsigned 32)

Path_B_Check timer reload value
This parameter contains the value of the Path_B_Check timer in microseconds.

(Unsigned 32)

Active_Port_Swap timer reload value
This parameter contains the value of the Active_Port_Swap timer in seconds.

(Unsigned 32)

Number of designated nodes
This parameter contains the number of nodes in designated node list.

(Unsigned 16)

Designated node list
This parameter contains the list of MAC addresses of designated nodes. It is applicable to beacon end nodes.

(Array of OctetString16)

Result (+)
This parameter indicates that the service request succeeded. The following fields are included in the response:

Node name
Source MAC address
Destination MAC address
Status

Result (-)
This parameter indicates that the service request failed and specifies error conditions, when applicable. The following fields are included in the response:

Node name
Source MAC address
Destination MAC address
Error info

8.7.4 Get node parameters service

Table 68 shows the parameters of the get node parameters service.

Table 68 – BRP get node parameters service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Node name	M	M(=)		
Source MAC address	M	M(=)		
Destination MAC address	M	M(=)		
Result (+)			S	S(=)
Node name			M	M(=)
Manufacturer	M	M(=)		
Version	M	M(=)		
Destination MAC address			M	M(=)
Node type			M	M(=)
Beacon timer reload value			C	C(=)
No_Beacon timer reload value			C	C(=)
Path_A_Check timer reload value			M	M(=)
Path_B_Check timer reload value			M	M(=)
Active_Port_Swap timer reload value			M	M(=)
Number of designated nodes			C	C(=)
Designated node list			C	C(=)

Parameter name	Req	Ind	Rsp	Cnf
VLAN ID			**C**	C(=)
Result (-)			**S**	**S(=)**
Node name			**M**	**M(=)**
Source MAC address			**M**	**M(=)**
Destination MAC address			**M**	**M(=)**
Error info			**M**	**M(=)**

Argument
The argument conveys the parameters of the service request. There are no specific parameters for this service.

Result (+)
This parameter indicates that the service request succeeded. The following fields are included in the response.

Manufacturer
This parameter contains the name of the manufacturer
(VisibleString255)

Version
This parameter contains the version of the BRP. Future versions of the BRP shall be downward compatible with the version specified in this clause. A BRP end node shall accept packets from end nodes supporting BRP versions lower than the one it supports. A BRP end node shall drop unknown packets and shall ignore extended payload contents in known packets from end nodes supporting BRP versions higher than the one it supports.
(Unsigned32)

Beacon timer reload value
This parameter contains the value of the beacon timer in microseconds.
(Unsigned 32)

No_Beacon timer reload value
This parameter contains the value of the No_Beacon timer in microseconds.

(Unsigned 32)

Path_A_Check timer reload value
This parameter contains the value of the Path_A_Check timer in microseconds.

(Unsigned 32)

Path_B_Check timer reload value
This parameter contains the value of the Path_B_Check timer in microseconds.

(Unsigned 32)

Active_Port_Swap timer reload value
This parameter contains the value of the Active_Port_Swap timer in seconds.

(Unsigned 32)

Number of designated nodes
This parameter contains the number of nodes in designated node list.

(Unsigned 16)

Designated node list
This parameter contains the list of MAC addresses of designated nodes. It is applicable to beacon end nodes.

(Array of OctetString16)

Result (-)
This parameter indicates that the service request failed and specifies error conditions, when applicable. The following fields are included in the response:

Node name
Source MAC address
Destination MAC address
Error info

8.7.5 Add node receive parameters service

Table 69 shows the parameters of the add node receive parameters service.

Table 69 – BRP add node receive parameters service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Node name	M	M(=)		
Source MAC address	M	M(=)		
Destination MAC address	M	M(=)		
Transmit node MAC address	M	M(=)		
Node receive timeout	M	M(=)		
Result (+)			S	S(=)
Node name			M	M(=)
Source MAC address			M	M(=)
Destination MAC address			M	M(=)
Status			M	M(=)
Result (-)			S	S(=)
Node name			M	M(=)
Source MAC address			M	M(=)
Destination MAC address			M	M(=)
Error info			M	M(=)

Argument
The argument conveys the parameters of the service request.

Transmit node MAC address
This parameter contains MAC address of transmit node of interest.

(VisibleString16)

Node Receive Timeout
This parameter contains associated node receive timeout in microseconds.

(Unsigned 32)

Result (+)
This parameter indicates that the service request succeeded. The following fields are included in the response.

Node name
Source MAC address
Destination MAC address
Status

Result (-)
This parameter indicates that the service request failed and specifies error conditions, when applicable. The following fields are included in the response:

Node name
Source MAC address
Destination MAC address
Error info

8.7.6 Remove node receive parameters service

Table 70 shows the parameters of the remove node receive parameters service.

Table 70 – BRP remove node receive parameters service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Node name	M	M(=)		
Source MAC address	M	M(=)		
Destination MAC address	M	M(=)		
Transmit node MAC address	M	M(=)		
Result (+)			S	S(=)
Node name			M	M(=)
Source MAC address			M	M(=)
Destination MAC address			M	M(=)
Status			M	M(=)
Result (-)			S	S(=)
Node name			M	M(=)
Source MAC address			M	M(=)
Destination MAC address			M	M(=)
Error info			M	M(=)

Argument
The argument conveys the parameters of the service request.

Transmit node MAC address
This parameter contains MAC address of transmit node to be removed from Node_Receive timers.

(VisibleString16)

Result (+)
This parameter indicates that the service request succeeded. The following fields are included in the response.

Node name
Source MAC address
Destination MAC address
Status

Result (-)
This parameter indicates that the service request failed and specifies error conditions, when applicable. The following fields are included in the response.

Node name
Source MAC address
Destination MAC address
Error info

8.7.7 Get node status service

Table 71 shows the parameters of the get node status service.

Table 71 – BRP get node status service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Node name	M	M(=)		
Source MAC address	M	M(=)		
Destination MAC address	M	M(=)		
Result (+)			S	S(=)
Node name			M	M(=)
Source MAC address			M	M(=)
Destination MAC address			M	M(=)
Node type			M	M(=)
Node status			M	M(=)
Port A status			M	M(=)
Port B status			M	M(=)
Result (-)			S	S(=)
Node name			M	M(=)
Source MAC address			M	M(=)
Destination MAC address			M	M(=)
Error info			M	M(=)

Argument
The argument conveys the parameters of the service request. There are no specific parameters for this service.

Result (+)
This parameter indicates that the service request succeeded. The following fields are included in the response.

Node status
This parameter contains the value representing node status.

(OctetString16)