CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – HIGH AVAILABILITY AUTOMATION NETWORKS –

Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)

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 patent:

   Patent WO06053459 "Reception of redundant and non-redundant frames", owned by ABB Switzerland Ltd, Corporate Research, Segelhofstr 1K, 5405 Baden, Switzerland.

   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."

International Standard 62439-3 has been prepared by subcommittee 65C: Industrial Networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of the IEC 62439 series is based on the following documents:

CDV	Report on voting
65C/519/CDV	65C/547A/RVC

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

This International Standard is to be read in conjunction with IEC 62439-1:2009, *Industrial communication networks – High availability automation networks – Part 1: General concepts and calculation methods.*

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

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[1] 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 standard may be issued at a later date.

1 The National Committees are requested to note that for this publication the maintenance result date is 2013. Amendments can be submitted before this date in accordance with IEC procedures.

INTRODUCTION

The IEC 62439-3 standard of the series "high availability automation networks", specifying the HSR and PRP redundancy protocols, was adopted by TC57 WG10 as the redundancy method for demanding substation automation networks based on IEC 61850, introducing new requirements.

IEC 62439-3 is now referenced in IEC 61850-8-1 Ed. 2 and IEC 61850-9-2 Ed. 2 (to be published end of 2010). As more complex nodes and networks were put together, several changes to the standard became necessary, in particular:

- clarification of specifications to ensure interoperability
- slackening of the specifications to allow different implementations
- simplify implementations
- consideration of clock synchronization according to IEEE 1588
- introduction of test modes to simplify testing and support

The major changes are:

Since these changes are already being implemented in development projects, they are formalized in this Amendment ahead of the maintenance phase to secure interoperability.

INDUSTRIAL COMMUNICATION NETWORKS – HIGH AVAILABILITY AUTOMATION NETWORKS –

Part 3: Parallel Redundancy Protocol (PRP) and High availability Seamless Ring (HSR)

1 Scope

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

This part of the IEC 62439 series specifies two redundancy protocols based on the duplication of the LAN, resp. duplication of the transmitted information, designed to provide seamless recovery in case of single failure of an inter-switch link or switch in the network.

2 Normative references

The referenced documents indispensable for the application of this part of the IEC 62439 series are listed in IEC 62439-1.

3 Terms, definitions, abbreviations, acronyms, and conventions

3.1 Terms and definitions

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

3.1.1
extended frame
frame that has been extended by a Redundancy Control Trailer

3.1.2

interlink
link that connects two network hierarchies

3.1.3
RedBox
device allowing to attach single attached nodes to a redundant network

3.1.4

QuadBox
Quadruple port device connecting two peer HSR rings, which behaves as an HSR node in each ring and is able to filter the traffic and forward it from ring to ring.

3.1.5

HSR frame
frame that carries as Ethertype the HSR_ethertype

3.2 Abbreviations and acronyms

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

DANH Double attached node implementing HSR

DANP Double attached node implementing PRP

RCT Redundancy Check Tag

SRP Serial Redundancy Protocol

VDAN Virtual Doubly Attached Node (SAN as visible through a RedBox)

3.3 Conventions

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

4 Parallel Redundancy Protocol (PRP)

4.1 PRP principle of operation

4.1.1 PRP network topology

This redundancy protocol implements redundancy in the devices, through doubly attached nodes operating according to PRP (DANPs).

An DANP is attached to two independent LANs of similar topology, named LAN_A and LAN_B, which operate in parallel. A source DANP sends the same frame over both LANs and a destination DANP receives it from both LANs within a certain time, consumes the first frame and discards the duplicate.

Figure 1 shows a redundant network consisting of two switched LANs, which can have any topology, e.g. tree, ring or meshed.

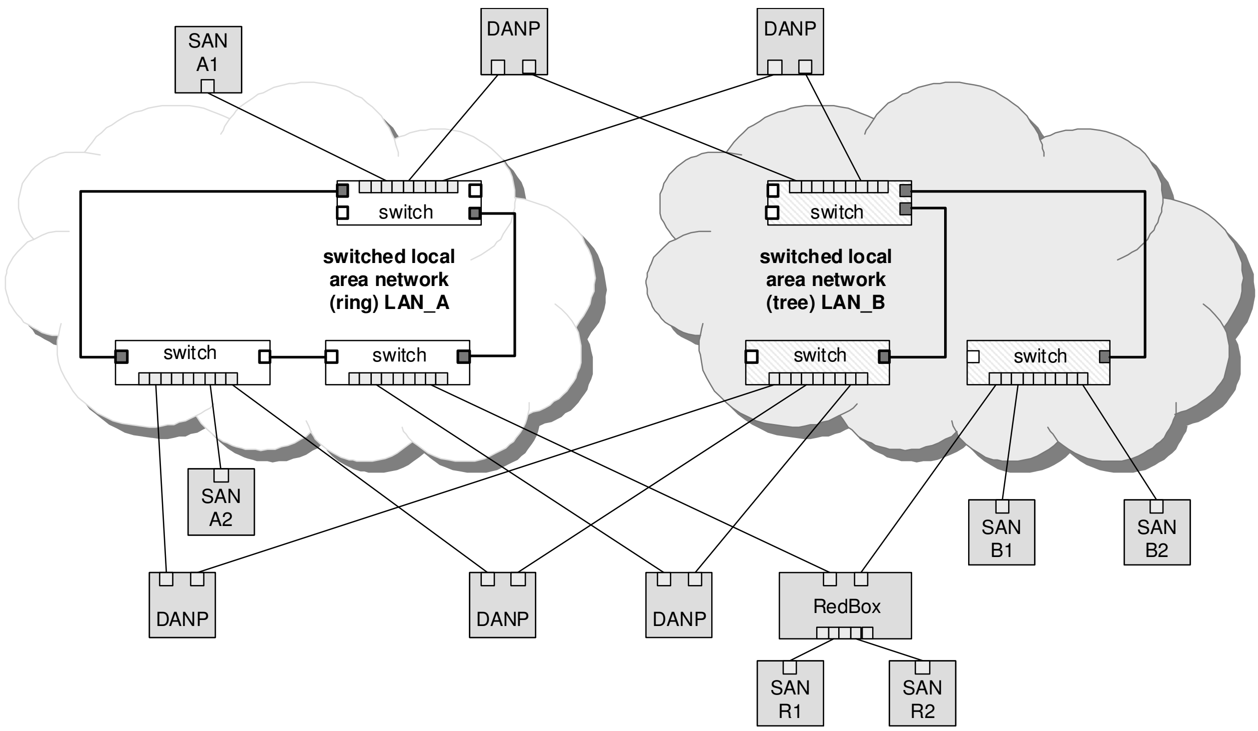

Figure 1 – PRP example of general redundant network

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, especially if one of the networks reconfigures itself, e.g. using RSTP, to overcome an internal failure.

The two LANs follow configuration rules that allow the network management protocols such as Address Resolution Protocol (ARP) to operate correctly.

The two LANs have no connection between them and 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 document provide guidance to the installer to achieve fail-independence.

4.1.2 PRP LANs with linear or bus topology

As an example of a simpler configuration, Figure 2 draws a PRP network as two LANs in linear topology, which may also be a bus topology.

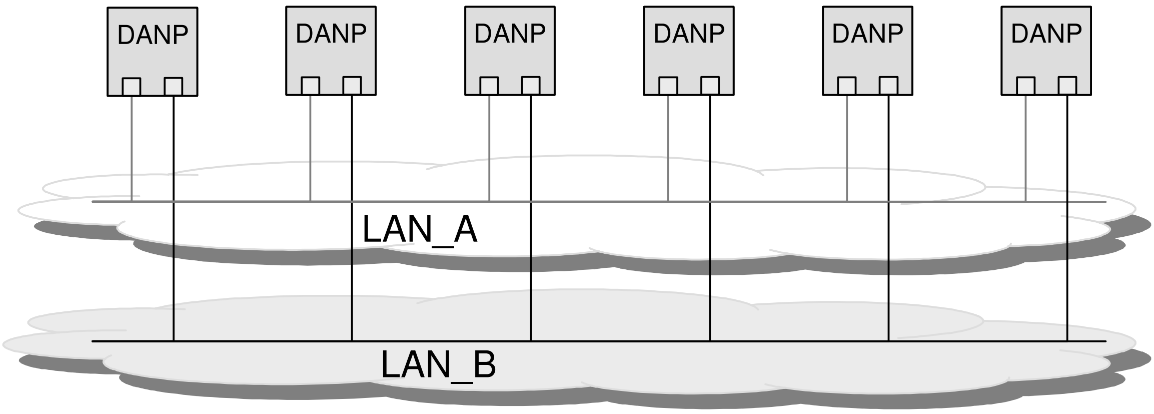

Figure 2 – PRP example of redundant network as two LANs (bus topology)

4.1.3 PRP LANs with ring topology

The two LANs can have a ring topology, as Figure 3 shows.

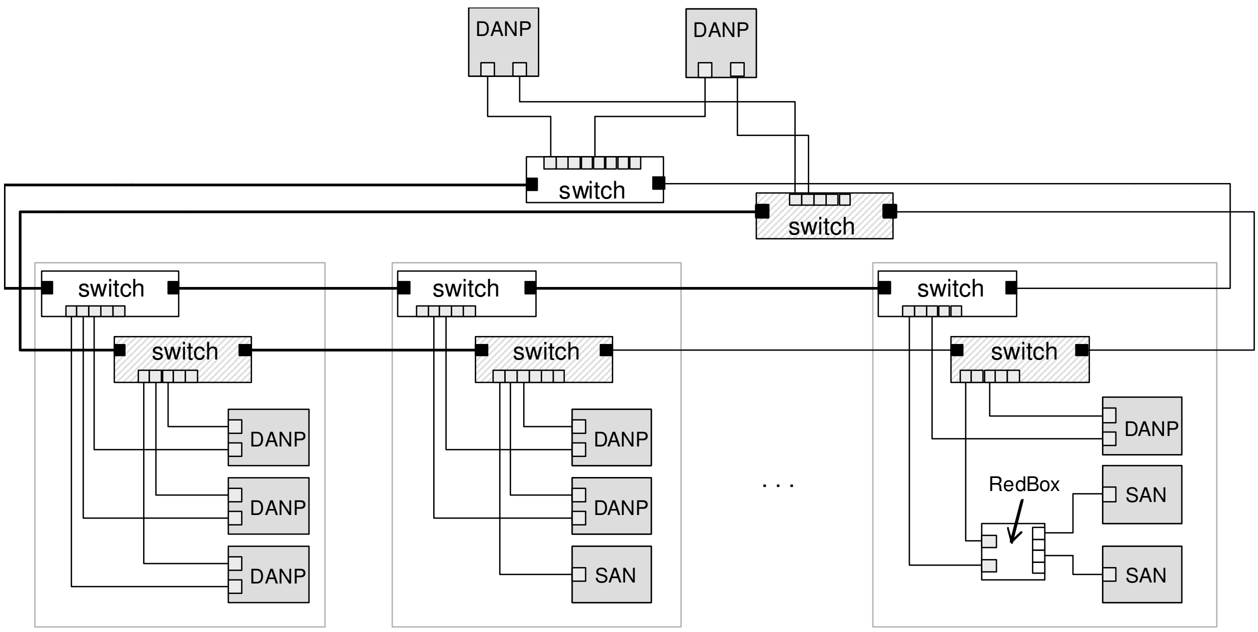

Figure 3 – PRP example of redundant ring with SANs and DANPs

4.1.4 DANP 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 Link Redundancy Entity (LRE), as Figure 4 shows.

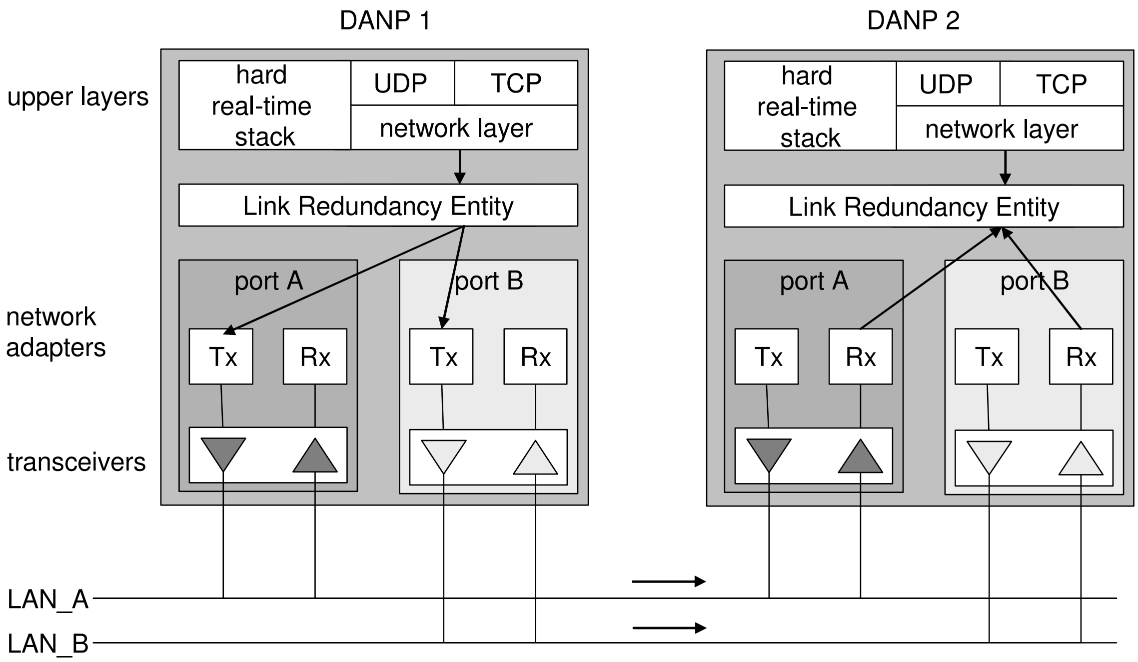

Figure 4 – PRP with two DANPs communicating

The Link Redundancy Entity (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.

4.1.5 PRP attachment of singly attached nodes

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

- SANs can be attached directly to one LAN only. SANs can only communicate with other SANs on the same LAN. For instance, in Figure 1, 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 RedBox (redundancy box) to both LANs, as Figure 1 shows for R1 and R2 (see also 4.1.9). Such SANs can communicate with all SANs, for instance SAN A1 and SAN R1 can communicate.

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

In some applications, only availability-critical devices need a double attachment, for instance the operator workplaces, while the majority of the devices are SANs. Taking advantage of the basic infrastructure of PRP, a DANP can be attached to two different switches of the same LAN (e.g. 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, where there is no forwarding of traffic between the ports. These abilities are optional and not detailed in this International Standard. The supported mode is specified in the PICS (see 6).

4.1.6 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 ISO/IEC 8802-3 (IEEE 802.3) padding. Conversely, DANPs understand the traffic generated by SANs, since these do not append a 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.

4.1.7 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 the same 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.

Delete the last paragraph.

4.1.8 Implication on configuration

Since the same frame can come from the two ports with significant time difference, the period of cyclic time-critical data must be chosen so that it considers the difference between worst case and best case path latency between publisher and subscriber.

4.1.9 Transition to non-redundant networks

The mechanism of duplicate rejection can be implemented by the RedBox that does the transition between a SAN and the doubled LANs, as Figure 5 shows. The RedBox mimics the SANs connected behind it (called VDA or virtual DANs) and multicasts supervision frames on their behalf, appending its own information. The RedBox is itself a DANP and has its own IP address for management purposes, but it may also perform application functions.

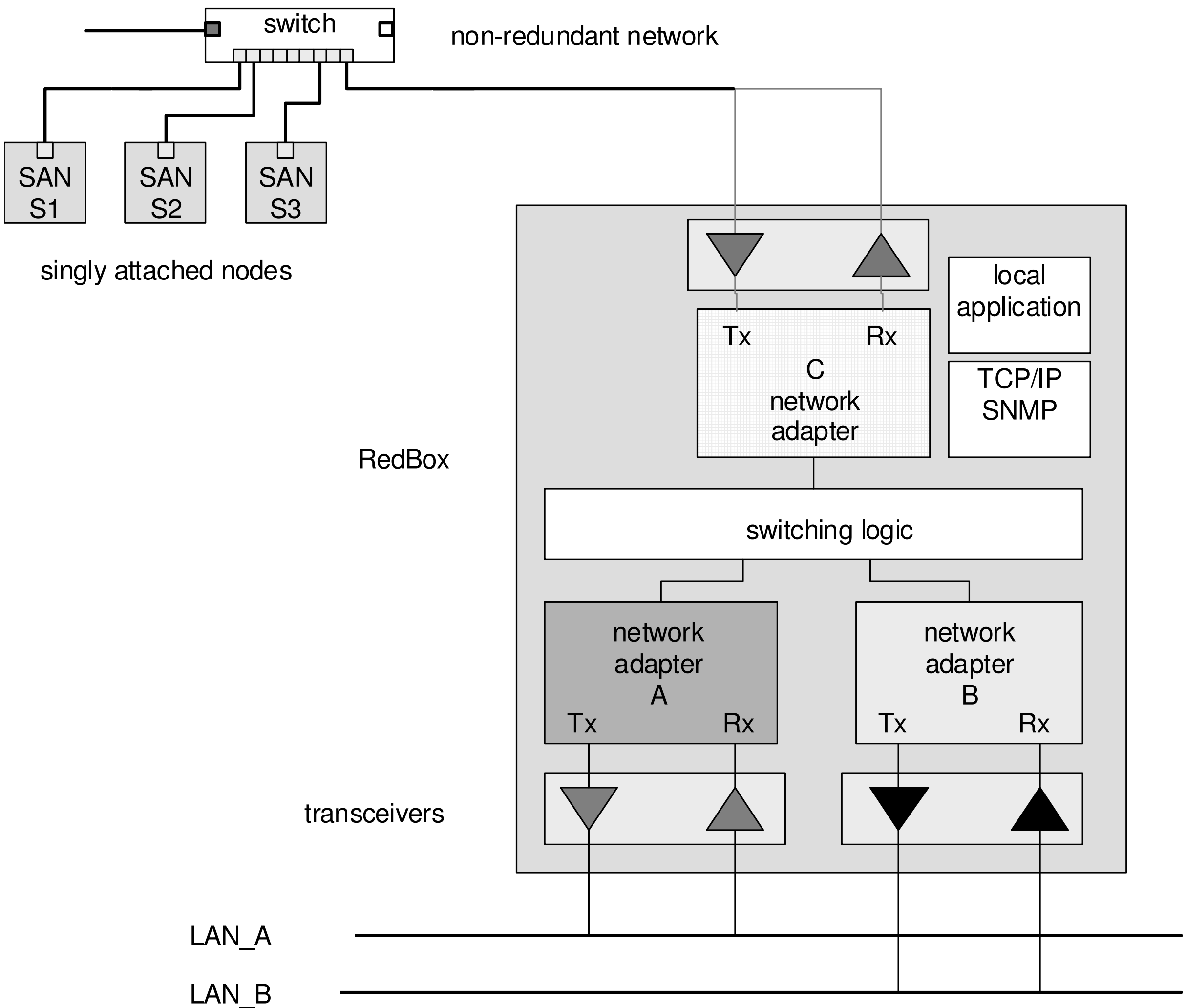

Figure 5 – PRP RedBox, transition from single to double LAN.

4.1.10 Duplicate handling

4.1.10.1 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.

4.1.10.2 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 International Standard, but it is not excluded.

4.1.10.3 Duplicate discard in the link layer

4.1.10.3.1 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 Redundancy Control Trailer (RCT), which the LRE inserts into each frame that it receives from the upper layers before sending, as Figure 6 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).
d) For PRP-1, a 16-bit suffix

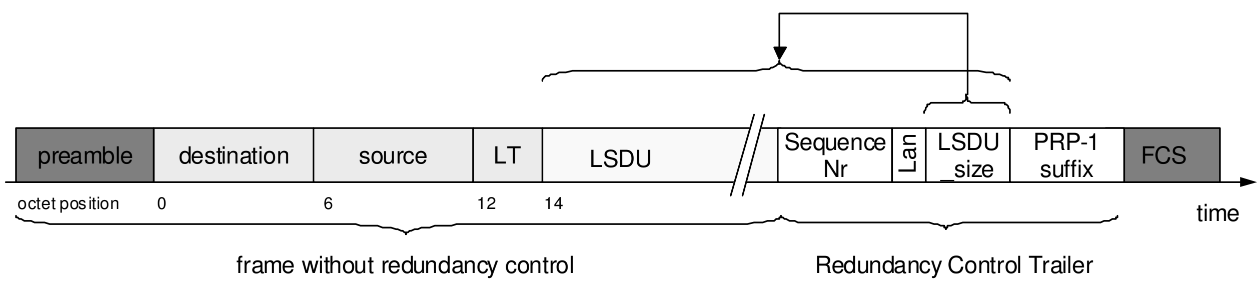

Figure 6 – PRP frame extended by an RCT

4.1.10.3.2 Use of SequenceNr

Each time a DANP sends a frame, it increases the sequence number and sends both (nearly identical) frames over both LANs.

There exist two options to increase the sequence number:

- PRP-0 increases the sequence number for each frame sent to a particular destination or multicast address, therefore a frame is uniquely identified by the source and destination MAC addresses and the sequence number. This scheme allows to keep track of communication relationships. PRP-0 is the original PRP specification and is to be considered as deprecated.
- PRP-1 increases the sequence number for each frame sent, regardless of its destination. This scheme allows to connect HSR and PRP networks with redundant connection points.

All nodes in a PRP network are assumed to be configured identically as PRP-0 or PRP-1. A correct configuration can be checked using the Supervisory frames.

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.

4.1.10.3.3 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.

4.1.10.3.4 Use of LSDU_size

To allow the receiver LRE to distinguish easily frames coming from DANP 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 VLAN tags may be added or removed during transit through a switch. To make the length field independent of VLAN tagging, only the LSDU and the RCT are considered in the LSDU_size, as Figure 7 shows.

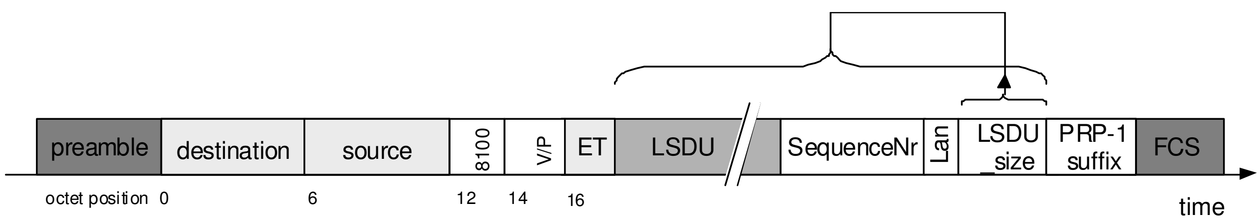

Figure 7 – PRP VLAN-tagged frame extended by an RCT

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 4.1.11), 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 8 shows.

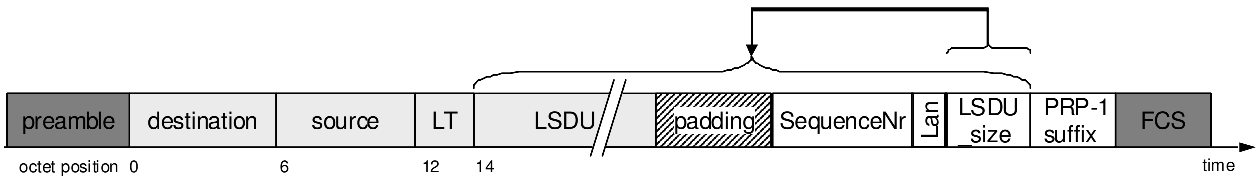

Figure 8 – PRP constructed, padded frame closed by an RCT

NOTE A VLAN-tagged frame can pass several switches which may remove or insert VLAN tags. If the sender observes the ISO/IEC 8802-3 (IEEE 802.3) rule to send a minimum frame size of 68 octets for a VLAN-tagged frame and of 64 for a VLAN-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.

4.1.10.3.5 Frame size restriction

Appending the RCT could generate oversize frames that exceed the maxValidSize foreseen by ISO/IEC 8802-3 (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-VLAN-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.

4.1.10.3.6 Discard algorithm

The algorithm for discarding duplicates is not specified. For instance, node tables, hash tables, fifos or sequence number tracking can be used. Whichever the algorithm, it must be designed such that it never rejects a legitimate frame, while occasional acceptance of a duplicate can be tolerated. Since the 16-bit sequence counter wraps around after 65536 frames and the sequence numbers are not necessarily contiguous, the algorithm must be able to forget entries older than a specified time EntryForgetTime, This also allows node reboot at any sequence number position.

The following Figure 9a shows the boundary conditions for discarding frames.

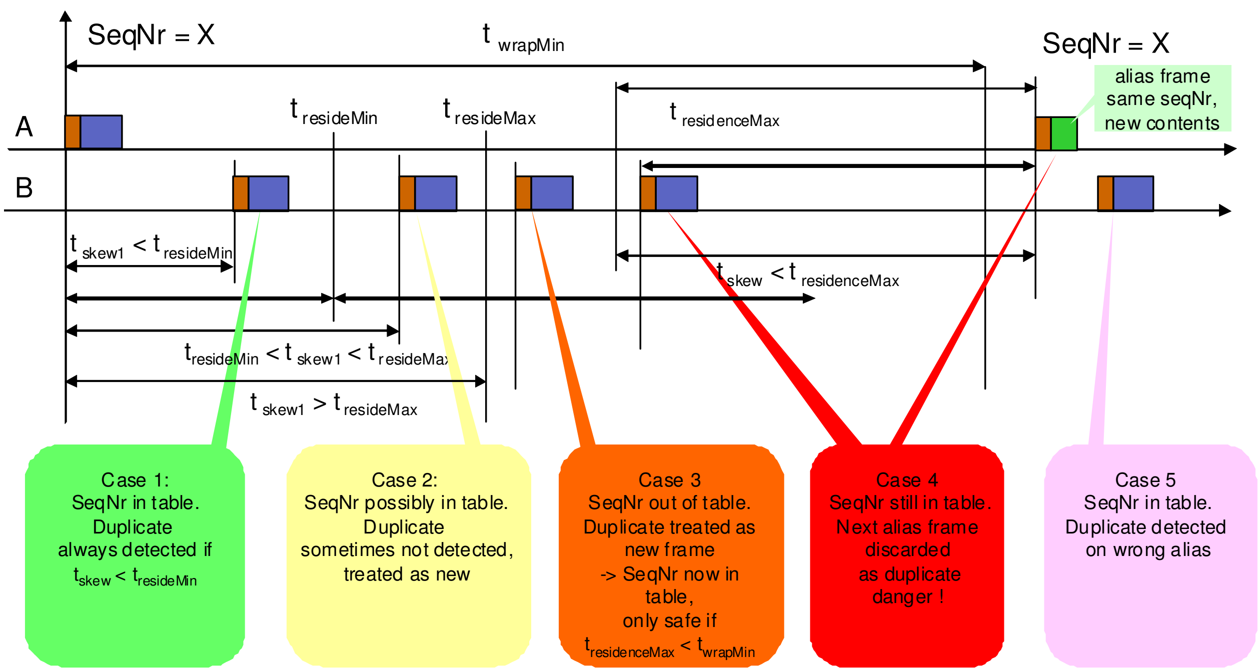

Figure 9a – Duplicate Discard algorithm boundaries

The time twrapMin is the minimum possible time between two repetitions of the same sequence number by legitimate frames after 65536 increments. This time is about 400 ms at the theoretical maximum frame repetition rate from the same source in a 100 Mbit/s network.

The time tskew is the time difference between the arrival of two copies of the same frame.

The time tresideMax is the maximum time an entry remains in the duplicate table, it must be smaller than EntryForgetTime.

A node that reboots must avoid aliases. This may be achieved by not sending frames during a time longer than EntryForgetTime on startup.

The following algorithm is an example how sequence number tracking could be applied to PRP-0.

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 9 shows for LAN_A. The same applies to ExpectedSeqB and StartSeqB on LAN_B.

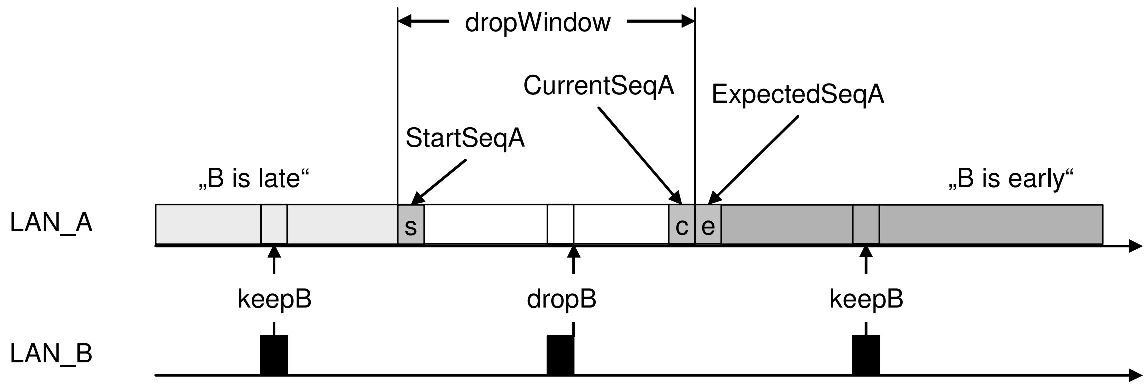

Figure 9 – PRP drop window on LAN_A

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 9), a frame from LAN_B whose sequence number CurrentSeqB fits into the drop window of A is discarded (dropB in Figure 9). In all other cases, the frame is kept and forwarded to the upper protocol layers (keepB in Figure 9).

Discarding the frame (dropB in Figure 9) 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 10 shows.

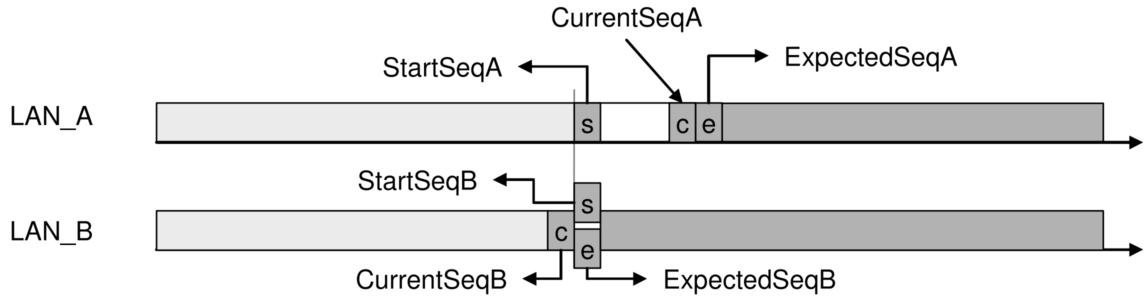

Figure 10 – PRP drop window reduction after a discard

In the situation of Figure 10, 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 11 shows.

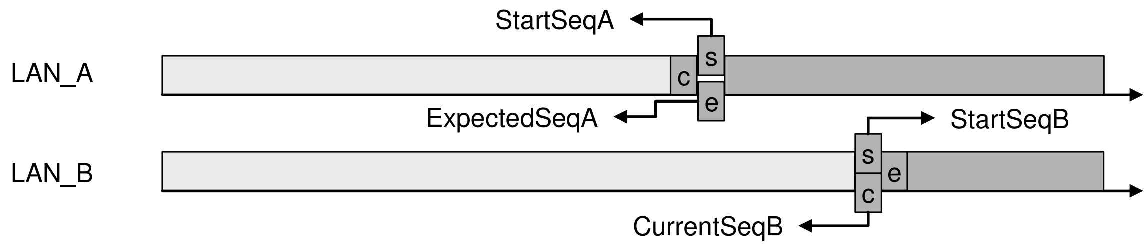

Figure 11 – PRP frame from LAN_B was not discarded

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 12 shows.

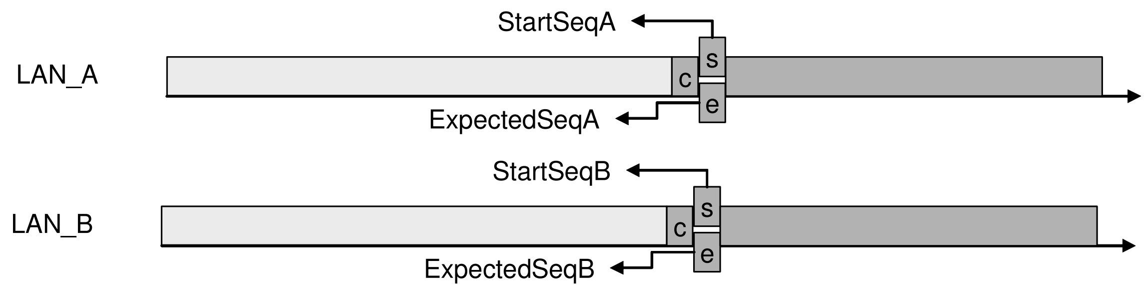

Figure 12 – PRP synchronized LANs

The sequence counter has 16 bits, which allows a drop window size of 32 768, 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 A discloses a pseudo-code for the duplicate discard algorithm.

4.1.10.3.7 Use of PRP-1 suffix

The PRP-1 suffix, in combination with the frame size, identifies PRP-1 frames. It allows to resolve situations when other protocols insert octets after the payload.

4.1.11 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.

4.1.12 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, senders and receivers should 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.

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.

4.1.13 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.

4.2 PRP protocol specifications

4.2.1 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.

4.2.1.1 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.

4.2.1.2 Labelling cables

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

4.2.1.3 Labelling switches

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

4.2.1.4 Independent operation

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

4.2.1.5 Configuration

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

All DANPs shall be configured with the same LifeCheckInterval.

4.2.2 MAC addresses

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

This address shall be unique in the network.

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).

4.2.3 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 4.2.7.6.

4.2.4 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.

4.2.5 Nodes

4.2.5.1 Node types

Doubly attached nodes according to the parallel redundancy protocol (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.

Singly Attached Nodes (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 RedBox.

4.2.5.2 Labelling connectors

This subclause 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 should be the upper connector and LAN_B the lower connector in its normal position.

When connectors are ordered horizontally, the left connector should 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.

4.2.6 Duplicate accept mode

4.2.6.1 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.

4.2.6.2 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.

4.2.7 Duplicate discard mode

4.2.7.1 Nodes table and duplicates tracking

Insert the following two new Subclauses before the 1st paragraph

4.2.7.1.1 Configuration modes

A DANP shall be configured either as PRP-0 or as PRP-1, depending on how the sequence number is incremented:

- In PRP-0, a node shall maintain a sequence counter SendSeq[i] for each destination (unicast or multicast) to which it sends a frame and increment it for each frame (pair) it sends to that specific destination, wrapping through zero.
- In PRP-1, a node shall maintain one sequence counter SendSeq and increment it for each frame (pair) it sends, wrapping through zero.

4.2.7.1.2 Monitoring data set and nodes table in PRP-0

Change the 1st paragraph and the numbered list as follows:

A node shall maintain a nodes 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 nodes table shall contain the following information for each unicast, multicast or broadcast address sent by node [i]:

e) SendSeq[i]
   a 16-bit sequence number used by this node for sending to that remote node or multicast or broadcast address ()
f) CntReceivedA[i] and CntReceivedB[i]
   for each adapter A and B, a 32-bit counter indicating the number of frames received over the adapter
g) CntErrWrongLanA[i] and CntErrWrongLanB[i]
   for each adapter A and B, a 32-bit counter indicating the number of mismatches on each adapter
h) TimeLastSeenA[i] and TimeLastSeenB[i]
   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.
i) SanA[i] and SanB[i]
   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 4.2.7.4.2).

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.

4.2.7.1.3 Monitoring data set and nodes table in PRP-1

A node shall maintain a monitoring data set consisting of the following elements:

a) SendSeq
   a 16-bit sequence number used by this node for sending
b) CntReceivedA and CntReceivedB
   for each adapter A and B, a 32-bit counter indicating the number of frames received over the adapter
c) CntErrWrongLanA and CntErrWrongLanB
   for each adapter A and B, a 32-bit counter indicating the number of mismatches on each adapter
d) AdapterActiveA and AdapterActiveB
   for each adapter A and B, the status of the communication link
e) An optional nodes table of all other nodes from which it receives supervisory frames.

NOTE 1 The nodes table allows to distinguish SAN from DANPs. A DANP without a nodes table will send all frames with a PRP-1 RCT over both interfaces. In application where broadcast traffic is the bulk, the impact on network load is small.

4.2.7.2 Redundancy Control Trailer (RCT)

The Redundancy Control Trailer (RCT) inserted into each DANP frame shall consist of four octets (PRP-0), resp. 6 octet (PRP-1), 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 (see 4.2.7.1).
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 ISO/IEC 8802-3 (IEEE 802.3) and IEEE 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, except the PRP-1 suffix.

d) A 16-bit PRP-1 suffix, set to 0x88FB

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

4.2.7.3 Sending (duplicate discard mode)

4.2.7.3.1 Frame size control

The sender shall have the ability to limit the LSDU size so that the complete frame, including the four-octet redundancy control trailer, 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 1 518 octets for VLAN untagged frames according to IEEE 802.3:2005.

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

4.2.7.3.2 Sending and nodes table for PRP-0

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 (PRP-0) 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 Redundancy Control Trailer (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, under the same conditions as 4.2.6.1.

4.2.7.3.3 Sending and nodes table for PRP-1

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

a) Increment the sequence number.

- if the destination address is registered neither as a SanA nor as SanB, increment the sequence number SendSeq, wrapping over through 0;

b) Send

- if the destination address is registered as either SanA or SanB, send the frame unchanged over the corresponding adapter;
- otherwise, append the Redundancy Control Trailer (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, under the same conditions as 4.2.6.1.

4.2.7.4 Receiver (duplicate discard mode)

4.2.7.4.1 Receiving and nodes table

On reception of a frame that has a destination MAC address different from the reserved addresses in IEEE 802.1Q -2004 Table 7-10 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, respectively 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.

4.2.7.4.2 Identification of frames associated with the duplicate discard mode

A PRP-0 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 Figure 6.

A PRP-1 receiver shall identify a duplicate candidate as a frame in which the 16-bit PRP-1 suffix can be identified, and whose 12 bits before the PRP-1 suffix match the physical size of the LSDU (exclusing the ethertype and the PRP-1 suffix).

For small frames that use padding, the receiver shall scan the frame backwards until it finds a matching 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.

4.2.7.4.3 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.

4.2.7.4.4 Duplicate discarding

A receiver can use any method to discard duplicates, provided that this method does not discard a frame sent as single or both frames of a pair, while it is permitted that in case of doubt, both frames of a pair can be passed to the higher protocol layers.

The duplicate discard method shall forget an entry after at most EntryForgetTime (see Table 2).

4.2.7.5 Cleanup of the nodes table

If it supports a 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 4.2.7.8).

NOTE It is sufficient to check the whole nodes table every NodeForgetTime for stale entries.

4.2.7.6 PRP_Supervision frame

4.2.7.6.1 Supervision frame for DANP

Each DANP shall multicast a PRP_Supervision frame over both its adapters with the format specified in Table 1 every LifeCheckInterval (see 4.2.7.8). This format shall also be used when the node is operating in duplicate accept mode.

Table 1 – PRP_Supervision frame with optional VLAN tag

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0						msb	U/L	I/G								
2	PRP_DestinationAddress = multicast (01-15-4E-00-01-XX)															
4									lsb							
6						msb	U/L	0								
8	PRP_SourceAddress (MAC address of the adapter)															
10									lsb							
12	ptid (0x8100 for VLAN)															
14	prio			cti	vlan_identifier											
16	SupEtherType															
18	SupPath				SupVersion < 64											
20	SupSequenceNumber (PRP-1)															
22	TLV.Type = 20 or 21								TLV.Length = 12							
24						msb	U/L	0								
26	MacAddressA (MAC address A of the DANP)															
28									lsb							
30						msb	U/L	0								
32	MacAddressB (MAC address B of the DANP, only for PRP-0)															
34									lsb							
36	TLV2.Type = 30								TLV2.Length = 6							
38						msb	U/L	0								
40	RedBoxMacAddress															
42									lsb							
44	TLV3.Type = 31								TLV3.Length = 6							
46						msb	U/L	0								
48	VdanMacAddress															
50									lsb							
	Padding to 64 octets (no VLAN) or to 68 octets (VLAN)															
58	SequenceNr															
60	Lan (0x1010 or 0x1011)						LSDU_size = 46									
62	Suffix for PRP-1															
64	FCS															
66																

4.2.7.6.2 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.

SupEtherType
Ethertype for the PRP protocol, shall be 0x88FB

SupPath
Path over which this frame was sent.

SupVersion
Indicates the protocol version, set to "0" (zero) for PRP-0 or set to "1" for PRP-1.
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. The version shall not exceed the value of 64, since the same beacon is used for HSR.

SupSequenceNumber
Sequence number incremented by 1 for each supervision frame sent, reused on HSR.

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.

TLV.Length
Indicates the length of the following MAC addresses.

MacAddress
MAC addresses used by each port. If the PRP_TLV.Length is 12, the MAC address of PortA is in the most significant 6 octets.

TLV2
This field shall be set to 0 if the source node is not a RedBox (see 4.2.7.6.3)

SequenceNr
Sequence number used for PRP traffic.

Lan
LAN identity over which this frame is sent.

LSDU_size
Size of the LSDU.

Suffix
PRP-1 suffix.

NOTE 1 Octets with offset 14 to 17 are inserted only if a VLAN tag according to IEEE 802.1Q is used.

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

NOTE 3 The SequenceNr, Lan and LSDU_Size are reused when the frame is passed to an HSR network.

4.2.7.6.3 PRP_Supervision frame for RedBox

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

TLV2.Type
Indicates the operation mode and shall have a value of 30 to indicate that the node is a RedBox.

TLV2.Length
Indicates the length of the following MAC address.

RedBoxMacAddress
MAC address of the RedBox that acts as proxy for the other device..

TLV3.Type
Indicates the operation mode and shall have a value of 31 to indicate that the node is a VDAN represented by the RedBox.

TLV3.Length
Indicates the length of the following MAC address.

VdanMacAddress
MAC address of the VDAN for which the RedBox acts as proxy.

The VDAN fields can be repeated up to the maximum frame size. If this is not sufficient, a RedBox can send subsequent supervisory frames until all VDAN have been announced. A TLV4 = 0 shall close the list of VdanMacAddresses, a TLV4 = 1 means that the list is continued in a next supervisory frame.

4.2.7.6.4 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.

4.2.7.6.5 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.

4.2.7.7 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 IEEE 802.1D 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 IEEE 802.1D Clause 8 in which its ports can take the root, alternate/backup or designated role, subject to the PICS PRP_RSTP or
- MRP, see IEC 62439-2, 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.

4.2.7.8 Constants

The constant parameters are shown in Table 2.

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

Table 2 – 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
EntryForgetTime	Time after which an entry is removed from the duplicate table	500 ms

4.3 PRP service specification

4.3.1 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 3 – PRP arguments

Argument	Definition	Data type
NodeName	Node name in the LRE	VisibleString32
ManufacturerName	Name of the LRE manufacturer	VisibleString255 (can be read only)
VersionName	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 if 5: LRE is (to be) configured as an HSR switching node if 6: LRE is (to be) configured as a RedBox in H mode if 7: LRE is (to be) configured as a RedBox in A mode if 8: LRE is (to be) configured as a RedBox in B mode	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
EntryForgetTime	Maximum time an entry should be considered	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 4.3.2

4.3.2 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 4.2.7.1).

Table 4 – Node Table attributes

Argument	Definition	Data Type
MacAddressA	MAC address of the source node (6 octets)	OctetString6
MacAddressB	Auxiliary MAC address of the source node (6 octets) 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
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

1 MacAddressB is not a key attribute.

4.3.3 PRP write

This service shall be used to write values to the LRE of a DANP to control the PRP. Table 5 shows the parameters of this service.

NOTE For the meaning of Req, Ind, Rsp, Cnf, M, U and S, refer to ISO/IEC 10164-1.

Table 5 – PRP write

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
NodeName	M	M(=)		
ManufacturerName	M	M(=)		
VersionName	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(=)		

Parameter name	Req	Ind	Rsp	Cnf
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 4.3.1.

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 7

4.3.4 PRP read

This service shall be used to read the current status of the LRE from a DANP. Table 6 shows the parameters of this service.

NOTE For the meaning of Req, Ind, Rsp, Cnf, M, U and S, refer to ISO/IEC 10164-1.

Table 6 – PRP read

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
none				
Result (+)			S	S(=)
NodeName			M	M(=)
ManufacturerName			M	M(=)
VersionName			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(=)

Parameter name	Req	Ind	Rsp	Cnf
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 4.3.1.

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 Clause 7).

5 High-availability Seamless Redundancy (HSR)

5.1 HSR objectives

Clause 5 describes the application of the PRP principles (Clause 4) to implement a High-availability Seamless Redundancy (HSR), retaining the PRP property of zero recovery time, applicable to any topology, in particular rings and rings of rings.

With respect to PRP, HSR allows to roughly halve the network infrastructure. With respect to rings based on IEEE 802.1D (RSTP), IEC 62439-2 (MRP), IEC 62439-6 (DRP) or IEC 62439-7 (RRP), the available network bandwidth for network traffic issomewhat reduced depending on the type of traffic. Nodes within the ring are restricted to be HSR-capable switching end nodes, thus avoiding the use of dedicated switches. Singly Attached Nodes (SANs) such as laptops or printers cannot be attached directly to the ring, but need attachment through a RedBox (redundancy box).

5.2 HSR principle of operation

5.2.1 Basic operation with a ring topology

As in PRP, a node has two ports operated in parallel; it is a DANH (Doubly Attached Node with HSR protocol).

A simple HSR network consists of doubly attached switching nodes, each having two ring ports, interconnected by full-duplex links, as shown in the example of Figure 13 (multicast) and Figure 14 (unicast) for a ring topology.

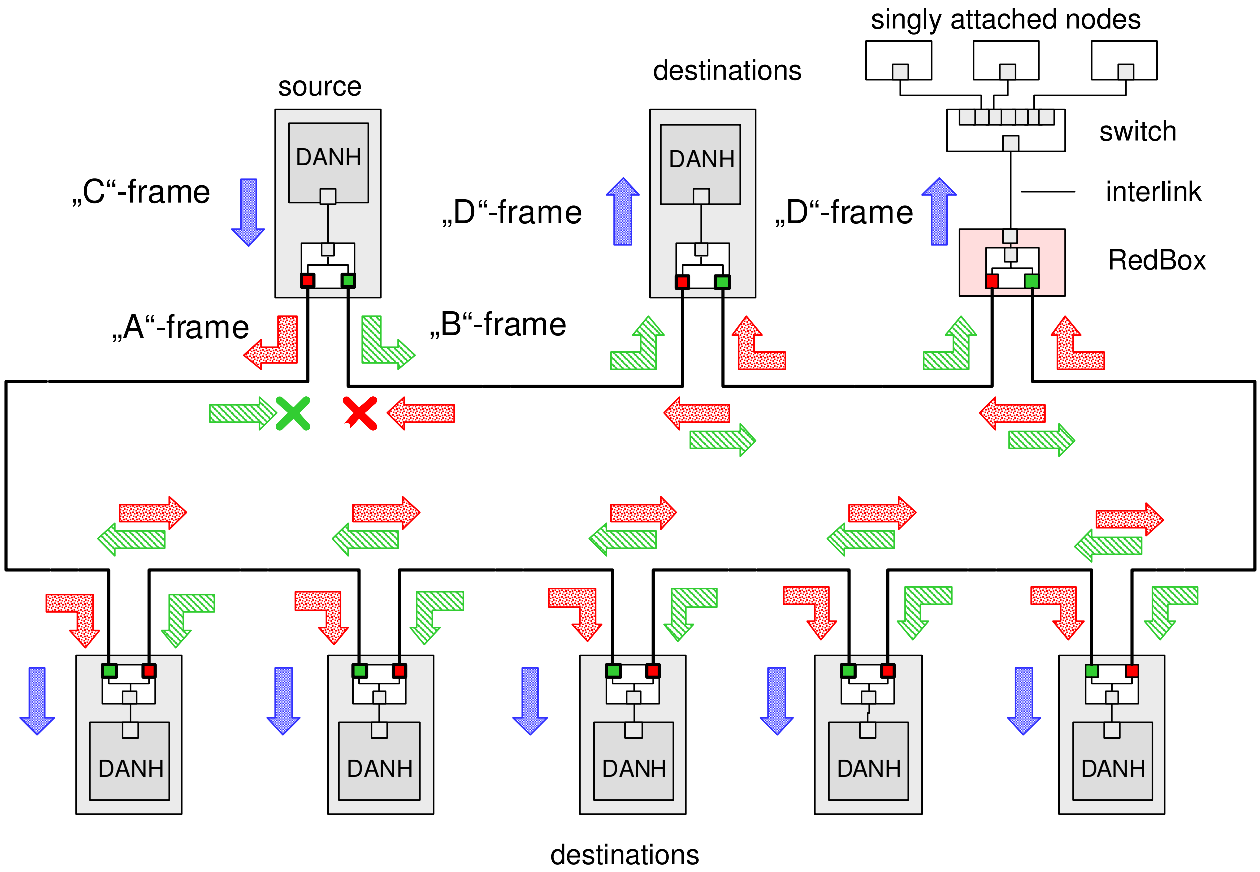

Figure 13 – HSR example of ring configuration for multicast traffic

Legend:
red, dotted arrows: “A” frames
green, cross-hatched arrows “B” frames
blue arrows: non-HSR frames exchanged between ring and host
cross: frame is removed from the ring by the next node.

A source DANH sends a frame passed from its upper layers (“C” frame), inserts an HSR tag to identify frame duplicates and sends a frame over each port (“A”-frame and “B”-frame).

A destination DANH receives, in the fault-free state, two identical frames from each port within a certain interval, removes the HSR tag of the first frame before passing it to its upper layers (“D”-frame) and discards any duplicate.

The nodes support the IEEE 802.1D bridge functionality and forward frames from one port to the other, except if they already sent the same frame in that same direction.

In particular, the node will not forward a frame that it injected into the ring.

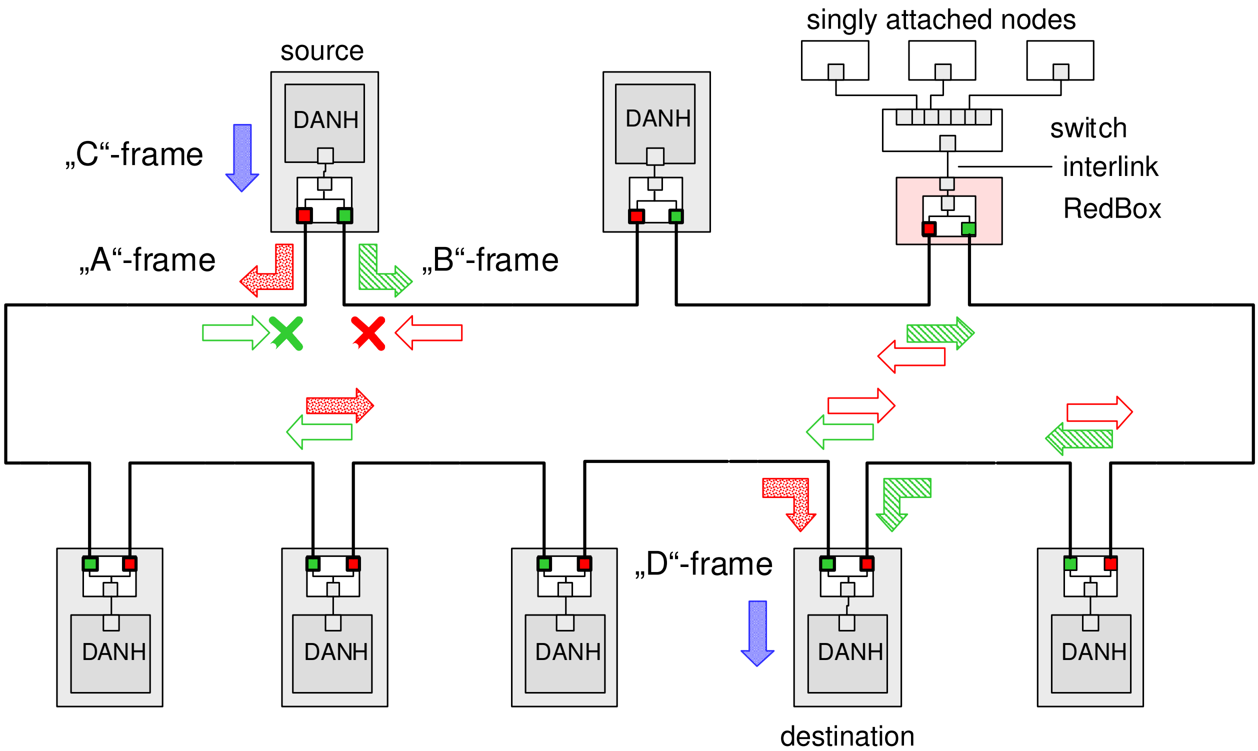

Figure 14 – HSR example of ring configuration for unicast traffic

Legend:
red dotted arrows: "A" frames
green cross-hatched arrows "B" frames
blue arrows: non-HSR frames exchanged between ring and host
cross: frame is removed from the ring by the next node

A destination node of a unicast frame does not forward a frame for which it is the only destination.

Frames circulating in the ring carry the HSR tag inserted by the source, which contains a sequence number. The doublet {source MAC address, sequence number} uniquely identifies copies of the same frame.

NOTE The time skew between two frames of a pair depends on the relative position of the receiving node and of the sending node. Assuming a worst case in which each node in the ring is transmitting at the same time its own frame with the largest size of 1 536 octets (maximum length supported by the Ethertype defined in IEEE 802.2), each node could introduce 125 µs of delay at 100 Mbit/s. With 50 nodes, the time skew may exceed 6 ms.

5.2.2 DANH node structure

Figure 15 shows a conceptual view of the structure of a DANH implemented in hardware, practical implementations can be different. The two HSR ports A and B and the device port C are connected by the LRE, which includes a switching matrix allowing to forward frames from one port to the other. The switching matrix allows cut-through switching. The LRE presents to the higher layers the same interface as a standard Ethernet transceiver would do.

The input circuit checks if this node is the destination of the frame and possibly does VLAN and multicast filtering to offload the processor. The duplicate discard is implemented in the output queues.

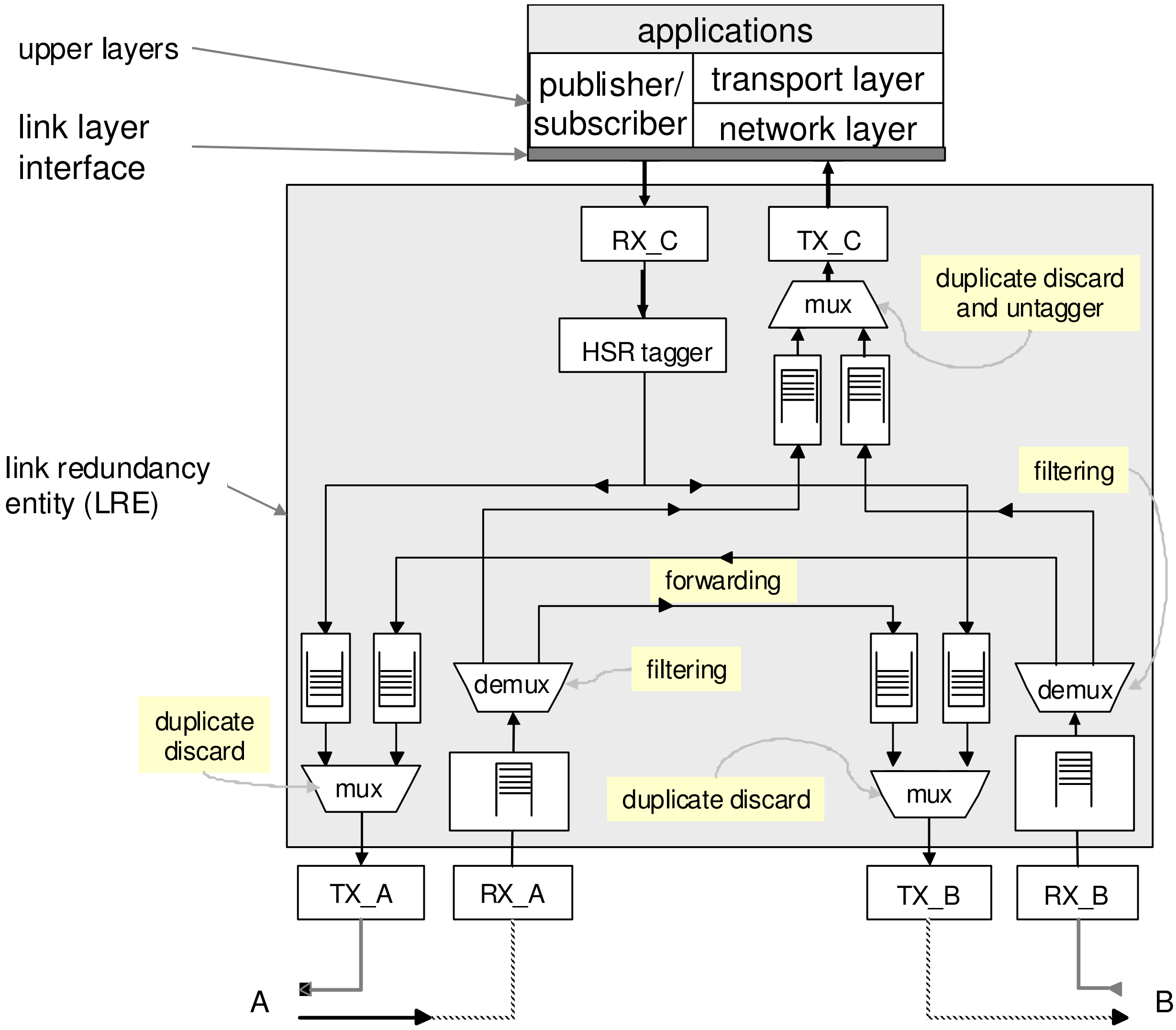

Figure 15 –HSR structure of a DANH

5.2.3 Topology

5.2.3.1 Attachment of singly attached nodes

Singly attached nodes (SAN), for instance maintenance laptops or printers cannot be inserted directly into the ring since they have only one port and cannot interpret the HSR tag in the frames. SANs communicate with ring devices through a RedBox (redundancy box) that acts as a proxy for the SANs attached to it, as shown in Figure 13 and Figure 14. The RedBox is detailed in 5.2.4.

Connecting non-HSR nodes to ring ports, breaking the ring, is allowed to enable configuration. Non-HSR trafic within the closed ring is supported in an optional mode.

5.2.3.2 Use of HSR with separate LANs

HSR nodes can be connected in the same way as PRP nodes. To this effect, an HSR node may be set to “no forwarding” (mode N). Unlike PRP, SANs cannot be attached directly to such a duplicated network unless they are able to interpret the HSR tag. The details of this operation mode are not covered in this version of the standard.

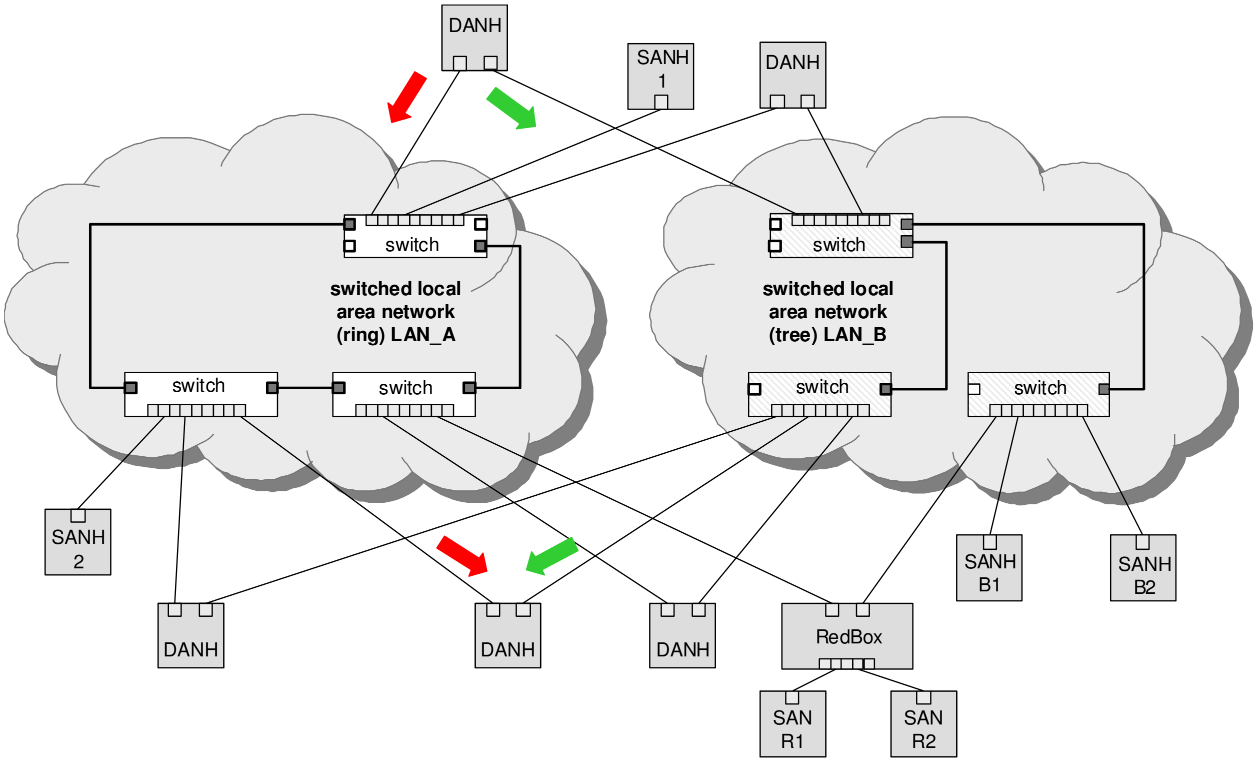

Figure 16 – HSR example of topology using two independent networks

5.2.3.3 Peer coupling of rings

Two HSR rings may be connected by quadruple port devices with forwarding capabilities, called QuadBoxes, as Figure 17 shows. This is advantageous when the traffic flow exceeds the capabilities of a single ring. However, transmission delays from end to end are not improved.

Although one QuadBox is sufficient to conduct the traffic in the fault-free state of the network, two QuadBoxes are used to prevent a single point of failure.

A Quadbox forwards frames over each ring as any HSR node, and passes the frames unchanged to the other ring, except if the frame can be identified as a frame not to be forwarded to the other ring. To this effect, a QuadBox is expected to filter traffic based for instance on multicast filtering or on VLAN filtering. There is no learning of MAC addresses in a QuadBox, though, since the learning of MAC addresses on specific ports of a QuadBox device could lead to a short break in communication if the QuadBox that has learned an address and is forwarding network traffic fails.

With QuadBoxes realized as single physical entities, the two interconnected rings share the same redundancy domain concerning fault tolerance. If one QuadBox breaks down, both interconnected rings are in a degraded state and cannot tolerate a further fault.

Therefore, constructing QuadBoxes in the same way as a RedBox can help keep the redundancy independent. The QuadBox then consists of two devices connected by an interlink. For this reason, the RedBox specifications include the HSR connection.

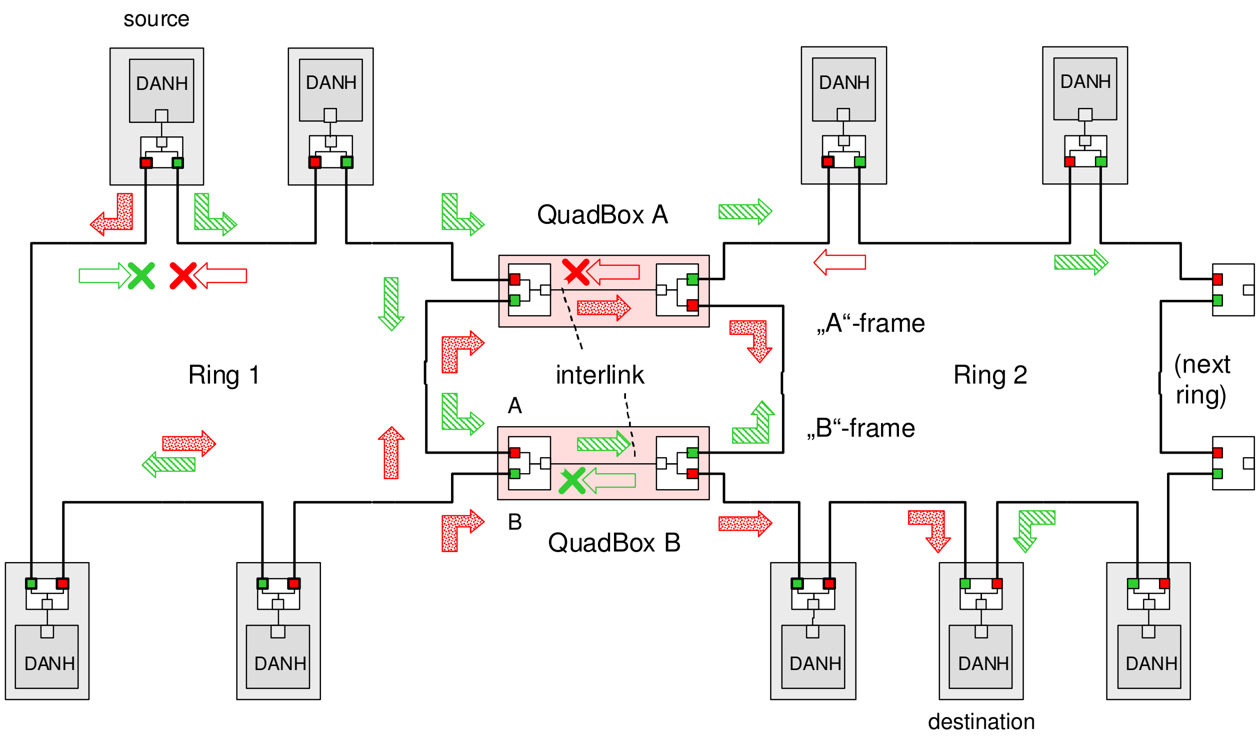

Figure 17 – HSR example of peer coupling of two rings

The presence of two QuadBoxes on the same ring causes that two copies of the same frame are transferred from the first ring to the second, each generating other two copies.

This does not cause four frames to circulate on the second ring, since, when a copy from a first QuadBox reaches the second QuadBox on the same second ring, the second QuadBox will not forward it if it already sent a copy that came from its interlink. Conversely, if the second QuadBox did not yet receive a copy from its interlink, it will forward the frame, but not the copy that comes later from the interlink.

When a QuadBox receives a frame that it itself injected into the ring or a frame that the other QuadBox inserted into the ring, it forwards it to the interlink and to its other port if it did not already sent a copy. This duplicate will be discarded at the other end of the interlink. This scheme may cause some additional traffic on the interlink, but it allows to simplify the design of the logic.

NOTE The maximum time skew between two frames of a pair is about the same as if all nodes were on the same ring.

5.2.3.4 Hierarchical ring topology

An HSR network may consist of rings connected by QuadBoxes, as Figure 18 shows.

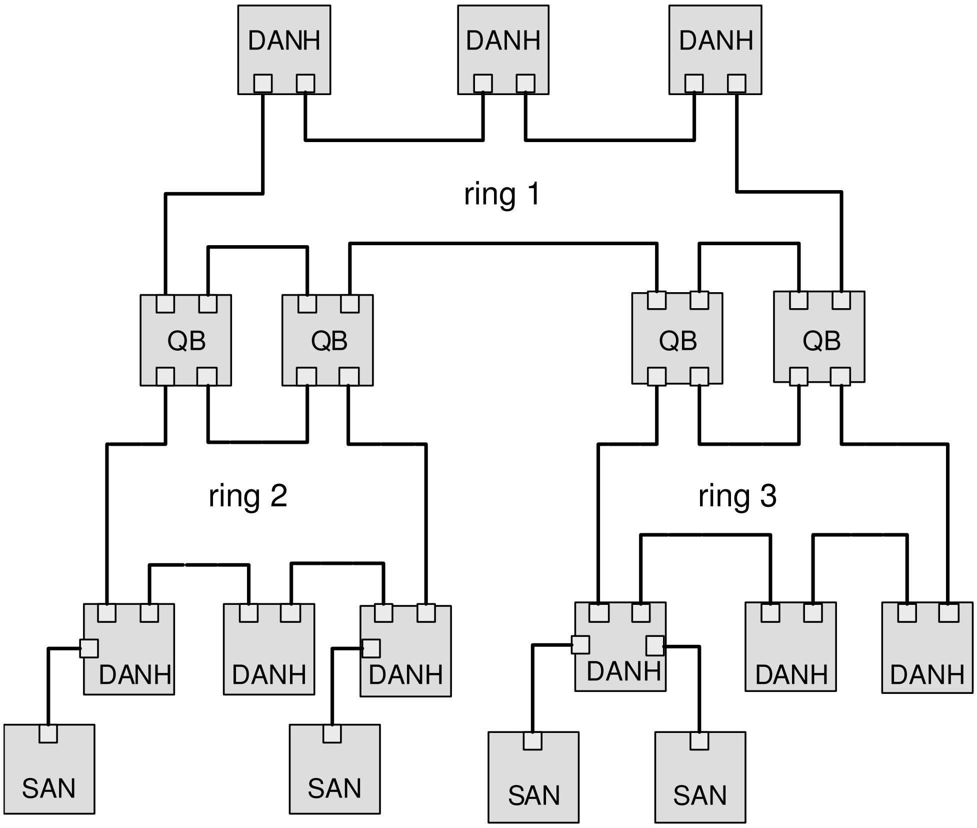

Figure 18 – HSR example of connected rings

Although a single QuadBox is sufficient to sustain the traffic, two independent QuadBoxes are needed to avoid a single point of failure.

Some SANs are connected directly to the DANH that performs the duty of a simplified RedBox.

5.2.3.5 Connection of an HSR ring to a PRP network

A HSR may be coupled to a PRP network through two RedBoxes, one for each LAN, as Figure 19 shows. In this case, the RedBoxes are configured to support PRP traffic on the interlink and HSR traffic on the ring ports.

The sequence number from the PRP-1 RCT is reused for the HSR tag and vice versa, to allow frame identification from one network to the other and to identify pairs and duplicates on the HSR ring, introduced by a twofold injection into the ring through the two HSR RedBoxes.

NOTE The coupling of PRP-0 to HSR is not intended.

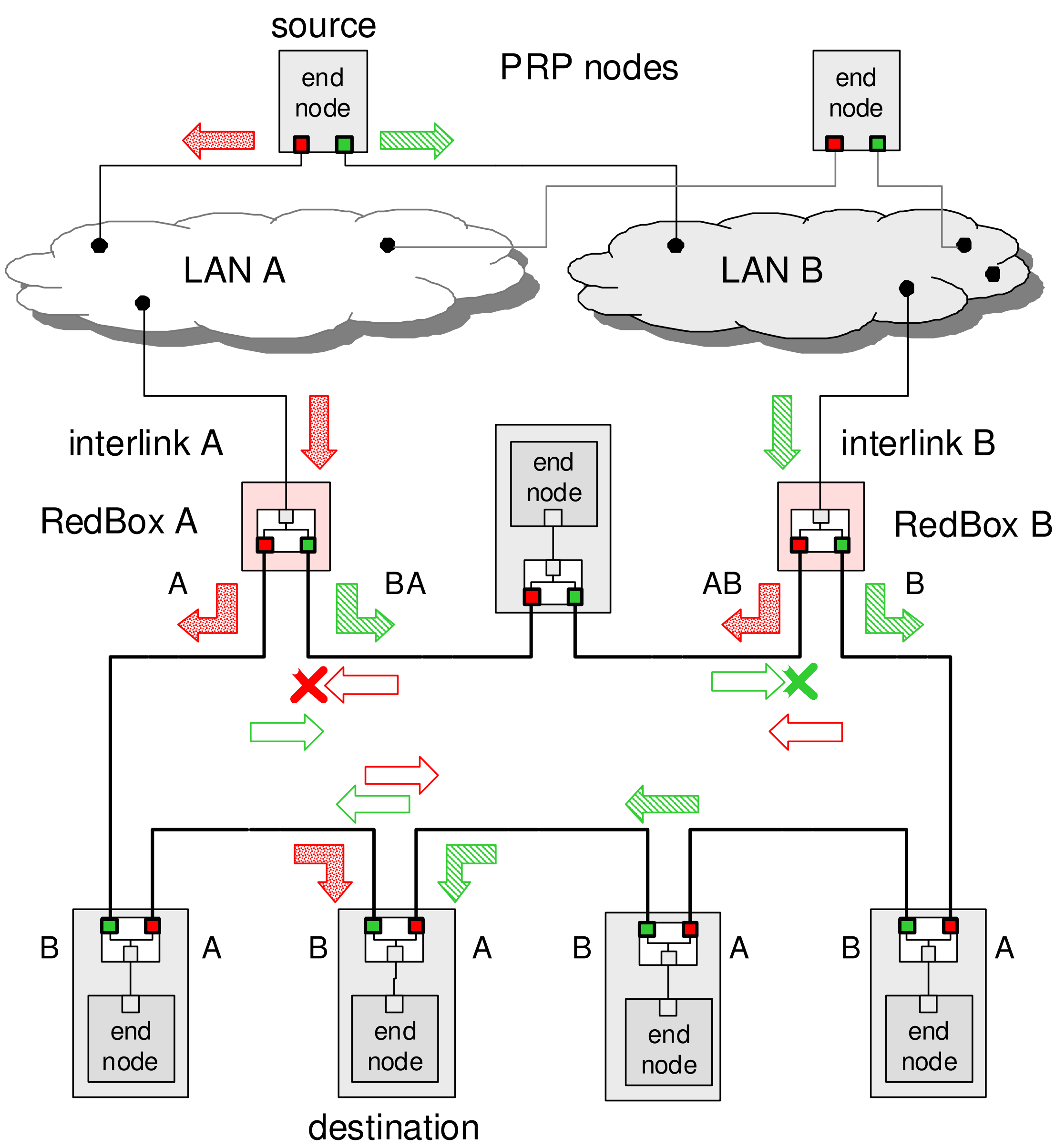

Figure 19 – HSR example of coupling two redundant PRP LANs to a ring

The HSR RedBoxes for connecting the ring to a PRP network operate identically to the HSR RedBoxes used to attach SANs described in 5.2.3.1, except that they are configured as RedBox "A" or RedBox "B" to accept PRP frames on their interlink.

In Figure 19, RedBox A and RedBox B would send the same frame (A and AB, respectively B and BA), but if a RedBox receives the frame before it could send it itself, it refrains from sending it.

In the example of Figure 19, RedBox A will not generate an "A" frame on behalf of LAN A if it previously received the same frame as "AB" from the ring, or conversely, RedBox "B" will generate an "AB" frame if it did not previously receive an "A" frame from the ring, which is the case whenever frame "A" is not a multicast frame.

Multicast frames or unicast frames without a receiver in the ring (void arrows in Figure 19) are removed by the RedBox that inserted them into the ring, if they originated from outside the ring.

Figure 20 shows the same coupling when the source is within the ring.

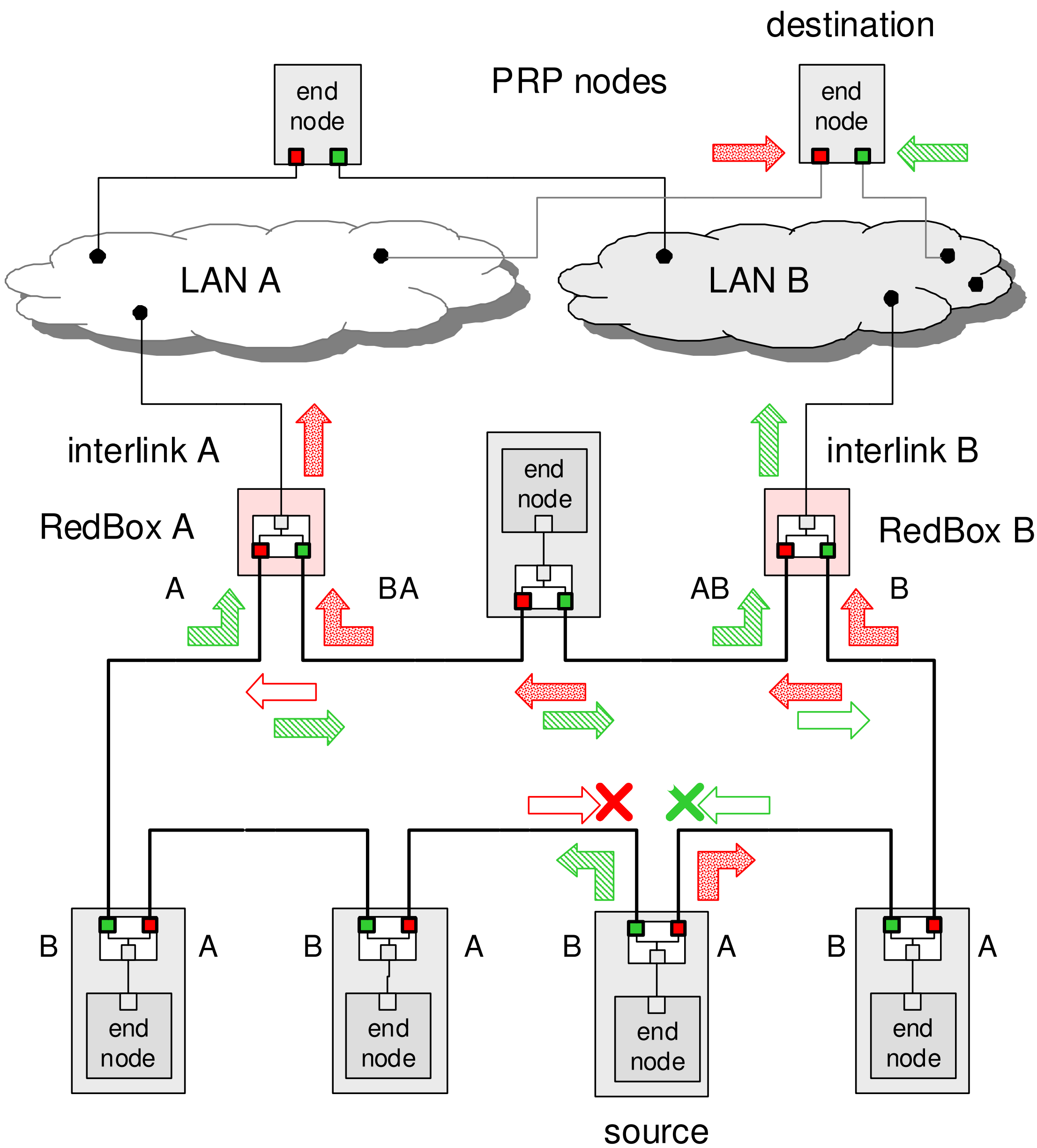

Figure 20 – HSR example of coupling from a ring node to redundant PRP LANs

It is necessary to configure the RedBox as a connexion to SAN or to PRP since the RedBox must insert the PRP trailer. However, letting the RedBox operate always in PRP mode does not harm, since the PRP trailer is invisible to the SANs.

5.2.3.6 Meshed topology

HSR allows any kind of meshing, and provides redundancy as long as the structure is free from single point of failure. For instance, Figure 21 shows for a matrix arrangement of nodes. In this case, nodes have more than two ports operated in parallel that operate like the QuadBoxes. A frame received from one port is forwarded to all other ports except the one that received it, and each port forwards the frame unless it already sent a duplicate.

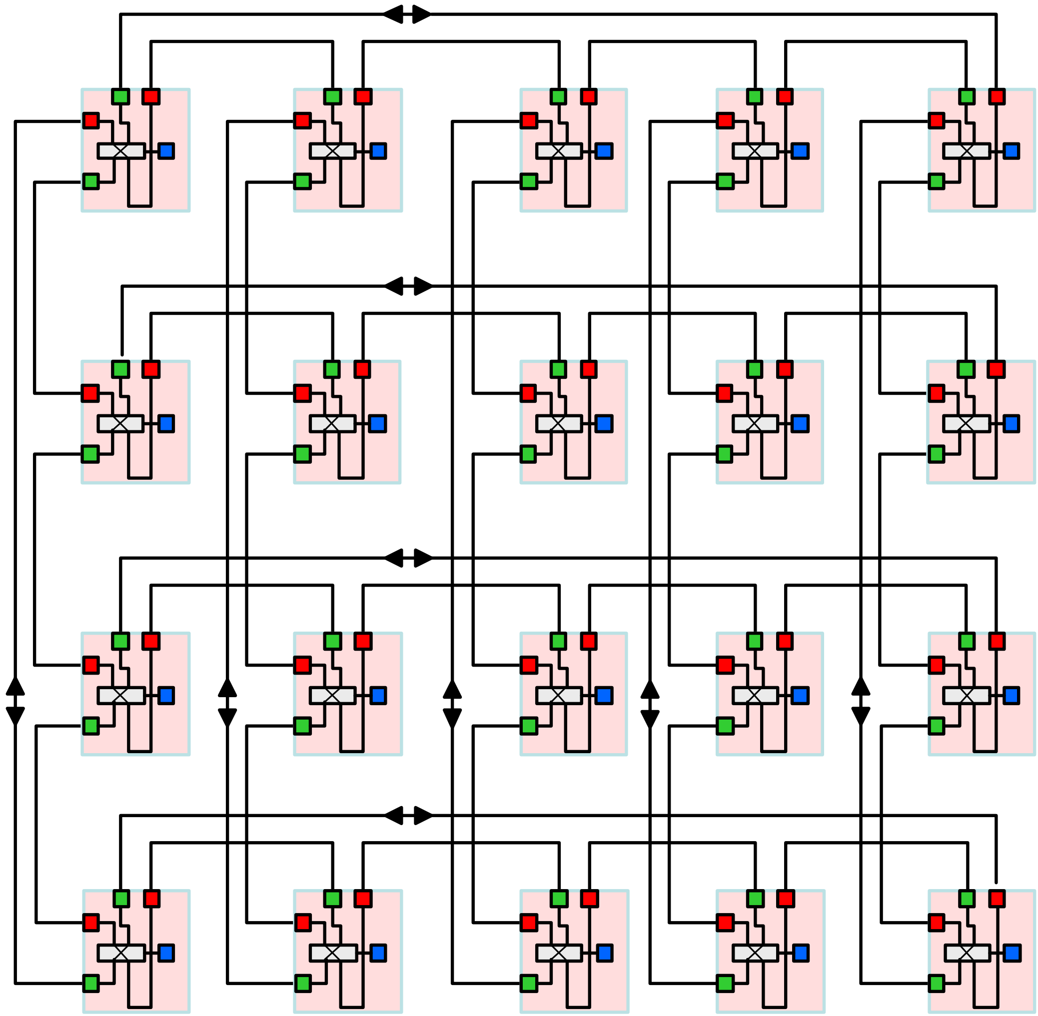

Figure 21 – HSR example of meshed topology

5.2.3.7 Connection of several PRP networks to an HSR ring

Up to 6 PRP networks can be connected to one ring. To this effect, each RedBox must have a different path identity and substitute the “A”/”B” of the PRP Lan identifier by respectively by one of 2/3, 4/5, 6/7, 8/9, A/B, C/D for sending on the ring. Converserly, they should substitute the HSR path identifier by “A”/”B” for sending towards the PRP network.

5.2.3.8 Connection of one PRP networks to several HSR rings

A PRP network can be connected to any number of HSR rings. However, the rings may not be connected between themselves, neither by QuadBoxes nor by another PRP network. Connection of more than one PRP network to more than one HSR ring is not permitted

5.2.4 RedBox structure

Figure 22 shows the general structure of a RedBox:

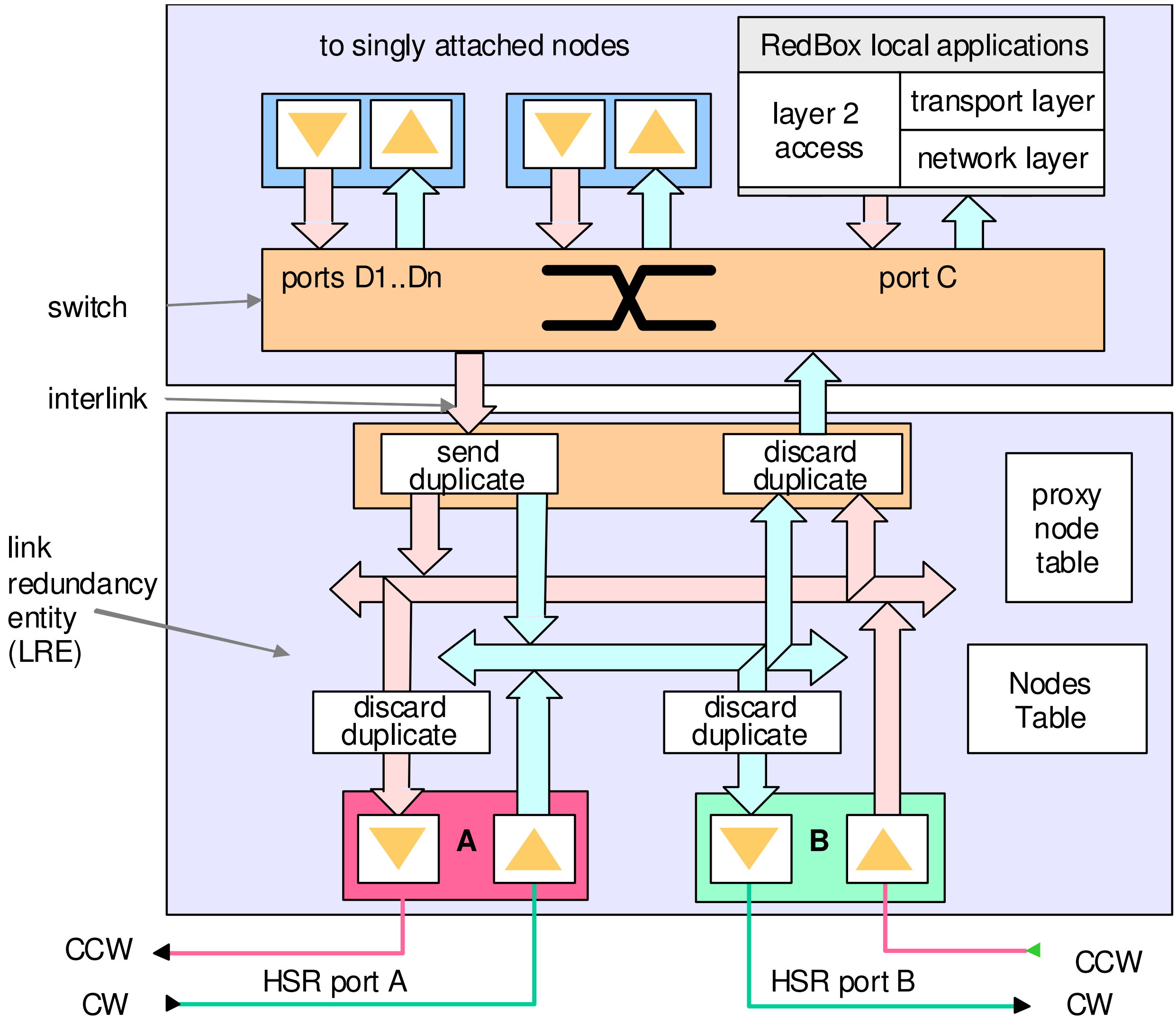

Figure 22 – HSR structure of a RedBox

The RedBox has a LRE that performs the duties of the HSR protocol, in particular:

- The RedBox forwards the frames received from one HSR port to the other HSR port, unless the frame was sent already.
- The RedBox receives frames addressed to its own upper protocols.
- The RedBox prefixes the frames sent by its own upper layers with the corresponding HSR tag before sending two copies over its HSR ports.

A RedBox may be operated in one of three modes: as a SAN connection, as a PRP connection or as an HSR connection (half of a QuadBox). Depending on the mode of operation, the frame handling at the interlink interface of the RedBox differs (see 5.4.2. and 5.4.4).

The switch in Figure 22 may be incorporated into the RedBox, so the interlink becomes an internal connection.

A simple RedBox is present in every node, since the LRE makes a transition to a single non-HSR host. Also, it is usual to have more than one host in a node, since a port for maintenance often exists.

5.3 HSR node specifications

5.3.1 HSR operation

5.3.1.1 Node properties

A DANH shall be operable in one of the following modes, which may be changed at runtime using management commands.

Mode H (mandatory, default mode): Forwarding with HSR Tag – in this mode, the DANH shall inserts the HSR tag on behalf of its host and forwards the ring traffic, except for frames sent by the node itself. Duplicate frames and frames where the node is the unicast destination (default mode)

Mode N (optional): No forwarding – in this mode, the DANH shall insert the HSR tag on behalf of its host, but does not forward the frame to its other ports, nevertheless rejecting duplicates.

Mode T (optional): Forwarding transparently all – in this mode the DANH shall remove the HSR tag before forwarding the frame to the other port and send a frame from the host to both ports, untagged and without discarding duplicates.

Mode M (optional): HSR-tagged and non HSR-tagged mixed mode – in this mode, the DANH shall insert the HSR tag depending on local criteria when injecting frames into the ring. HSR tagged frames from ring ports are handled according to Mode H. Non-HSR tagged frames from ring ports are handled according to IEEE 802.1D forwarding rules.

Mode U: Forwarding Unicast – same as Mode H, except that in this mode the DANH shall treat all traffic as multicast.

5.3.1.2 Host sequence number

An HSR node shall maintain a sequence number for each MAC address the host uses. The sequence number is initialized with 0.

NOTE The LRE is expected to detect the MAC address of the host by listening to the frames it receives from it. The host MAC address can also be configured into the LRE. Without knowledge of the host MAC address, the LRE forwards all HSR traffic to the host, treating unicast frames as multicast.

5.3.1.3 Host multicast address

A node shall have the ability to filter out at least MulticastFilterSize (see Table 8) multicast addresses or multicast address groups. If the table is not configured, the LRE shall forward all multicast frames to the host and to the other port, duplicates excluded.

5.3.1.4 Host unicast address

A node shall have the ability to communicate its MAC address(es) to the LRE. If the MAC address is not configured, the LRE shall forward all unicast frames to the host and to the other port, duplicates excluded.

5.3.1.5 Host reboot

When a node reboots, it shall refrain from sending frames over its HSR ports during a time NodeRebootTime (see Table 8) and preferibly start transmission by sending a supervision frame.

5.3.2 DANH receiving from its link layer interface

For each frame to send on behalf of its link layer interface, a source node shall:

If the port is in non-HSR mode	
	Do not modify the frame;
else (port is in HSR mode)	
	Insert the HSR tag with the sequence number of the host;
	Increment the sequence number, wrapping through 0
Duplicate the frame, enqueue it for sending into both HSR ports	

NOTE 1 Enqueuing means that the frame will be sent as soon as no former or higher-priority frames are in the queue and the medium is ready.

NOTE 2 Sending a non-HSR frame to both ports should not cause circulating frames since such frames will not be forwarded by the adjacent HSR node. This mechanism is intended to allow off-ring configuration of an HSR node through a normal PC.

5.3.3 DANH receiving from an HSR port

A node receiving a frame from one of its HSR ports shall:

If this frame is not HSR-tagged:			
	If the frame is a link local traffic (IEEE 802.1D Table 7-10)		
		Consume the frame and do not forward it.	
	Else if the node is in Mode M		
		Enqueue the unchanged frame for passing to its link layer interface.	
		Forward according to 802.1D forwarding rules	
	Else if the node is in Mode T		
		Set the port to non-HSR mode	
		Enqueue the unchanged frame for passing to its link layer interface.	
		Forward according to 802.1D forwarding rules	
	Else		
		Enqueue the unchanged frame for passing to its link layer interface.	
		Do not forward this frame	
Else (HSR-tagged frame):			
	Register the source in its node table (if implemented) as HSR node;		
	If this node is a destination:		
		If this is the first occurrence of the frame over the link layer interface:	
			Register the occurrence of that frame;
			Remove the HSR tag and pass the modified frame to its link layer interface.
		Else (this is not the first occurrence of the frame over the link layer interface):	
			Register the occurrence of that frame;
			Do not pass the frame to the link layer interface.
	Else (if this node is not a destination):		
		Do not pass the frame to the link layer interface.	
	If this node is not the only destination (multicast or unicast for another node or the node is in Mode U:		
		If this is the first occurrence of the frame over the second port:	
			Register the occurrence of that frame;
			Enqueue the unmodified frame for sending over the second port.
		Else (this is not the first occurrence of the frame over the second port):	
			Register the occurrence of that frame;
			Discard the frame.

	Else (If this node is the only (unicast) destination) and the node is not in Mode U :	
		Discard the frame.

NOTE 1 It is possible that more than one duplicate arrives, especially when rings are coupled.

5.3.4 DANH forwarding rules

A node shall not send over a port a frame that is a duplicate of a frame previously sent over that port in that same direction. This can also be seen in the behavioural description in 5.3.3.

A node that detects on the base of the signal quality supervision that the frame is damaged or truncated, shall not forward it. However, if the node operating in cut-through already started forwarding and then detects that the frame is damaged or truncated, it shall append the error sequence foreseen in IEEE 802.3 (2002) 27.3.1.2.2 and then stop transmission of that frame.

If a previously connected port is disconnected from the network, a node shall purge the port's buffer so that it cannot send an obsolete frame, and only allow buffering when the port is reconnected.

NOTE 1 These rules remove circulating HSR frames and open the ring, in the same way as an RSTP or similar protocol. It applies to frames originally sourced by the node and to frames circulating in case a device is removed after having sent a frame, and the ring is closed again, for instance by a mechanical bridging device or when a DANH is powered down. In a ring of 50 nodes, there may be a delay of some 6 ms until a frame comes back to its originator, so this possibility must be cared for.

NOTE 2 These conditions enable a node to operate either in store-and-forward or in cut-through mode. Delaying the forwarding of a frame does not affect the worst-case ring delay.

NOTE 3 The duplicate discard method of PRP is not a preferred method for discarding duplicates in HSR, since HSR aims at preventing duplicates from circulating.

NOTE 4 The fact that the sequence numbers of the frames sent by one source are not monotonically increasing is not a reason for discarding the frame. This observation can however be used for supervision of the network.

NOTE 5 For cut-through operation, the node must wait approximately 5 µs at 100 Mbit/s until the HSR tag has been completely received and the node decided to forward or not. By contrast, store-and-forward takes at least 122 µs at 100 Mbit/s for the maximum size frame (1522 octets).

NOTE 6 Cleaning the transmit queues from entries coming from this node should not be necessary, since a transmit queue should empty itself whether the attached link is operational or not.

5.3.5 CoS

For the operation of HSR, priorities and VLANs are not required.

An HSR node is expected, as expressed in its PICS:

- To support at least 2 levels of priority according to IEEE 802.1D (IEEE 802.1p);
- To filter VLAN traffic according to IEEE 802.1Q;
- To filter multicast traffic;

5.3.6 Clock synchronization (informative)

In case IEC 61588 (IEEE 1588) is used, and due to the fact that clock synchronization frames can arrive on both ports with different delays, it is recommended to use transparent – ordinary clocks (hybrid clocks) together with one-step mode.

In case the IEEE 1588 profile according to IEEE D37-238 is used, the IEEE 1588 frames shall receive an HSR Tag, except for the PdelayRequest and PdelayResponse frames, which are only exchanged between peer nodes.

NOTE One-step clocks require on-the-fly modification of the clock correction, which is only practical when done in hardware.

5.3.7 Deterministic medium access

HSR does not specify the traffic control that has to be used for deterministic, real-time operation.

However, it is a recommended practice to buffer the hard real-time, high priority frame, wait until the clock reaches a certain time, the same in all devices with the same period, and let all these nodes send this traffic at that time, in order to leave sufficient contiguous free space for the non-real-time traffic.

5.4 HSR RedBox specifications

5.4.1 RedBox properties

A RedBox is a device with at least three ports, two of them being ring ports for the HSR protocol, the third port being connected to an interlink.

A RedBox shall behave as a DANH for all traffic for which it is the source and the destination, as described in clause 5.3.1 and its subclauses. This is due to the fact that a RedBox itself implements at least a management interface. In addition to the DANH behaviour, the behaviour of the RedBox Interlink is described in 5.4.2. Since the RedBox distinguishes, on reception from the HSR network, between its internal interfaces and its Interlink, the behaviour of a RedBox upon reception from an HSR network is described in 5.4.4.

A RedBox shall have a configurable path identity, that is composed of four bits, the three most significant bits indicating the PRP network to which the RedBox is connected and the least significant bit indicating the LAN identity, A (0) or B (1). The three most significant bits may not be 000 or 111.

A RedBox shall be configurable for one of three modes:

1) HSR-SAN: the traffic on the interlink is not HSR, not PRP

2) HSP-PRP: the traffic on the interlink is PRP-tagged as “A” or “B”

3) HSR-HSR the traffic on the interlink is HSR-tagged.

A RedBox shall behave as a DANH for all traffic for which it is the source or the destination.

NOTE 1 A RedBox is expected to have its own IP address, especially for configuration messages. It can be accessed over the interlink or over the HSR ports.

NOTE 2 The interlink can be an internal connection if the RedBox serves as switch at the same time.

5.4.2 RedBox receiving from interlink

When receiving a frame from its interlink port, a RedBox shall:

- If the frame carries a HSR tag:
  - If the RedBox operates in HSR-HSR mode
    - If the RedBox is a destination of the frame
      - If this is not the first occurrence of the frame at the link layer interface
        - Register the occurrence
        - Discard the frame
      - Else (If this is the first occurrence of the frame at the link layer interface)
        - Register the occurrence
        - Remove the HSR tag
        - Enqueue to the link layer interface of the RedBox
    - If the frame is to be injected into the ring (RedBox is not only destination, Multicast/VLAN is ok)
      - If this is not the first occurrence of the frame at each HSR port
        - Register the occurrence
        - Discard the frame (already sent over that port)
      - Else (If this is the first occurrence of the frame at each HSR port)
        - Enqueue the unmodified frame into each HSR port
  - Else (If the RedBox does not operate in HSR-HSR mode)
    - Discard the frame

- Else if the frame carries a PRP RCT
  - If the source MAC address is not already registered:
    - Create an entry in the proxy node table;
  - If the PRP tag does not correspond to the LanId of the RedBox “A” or “B”
    - Register the error;
    - Discard the frame
  - Else (If the PRP tag corresponds to the e LanId of the RedBox “A” or “B”)
    - If the PRP frame was already received
      - Register the occurrence
      - Discard the frame
    - Else (if the PRP frame was not already received)
      - Register the occurrence
      - If the RedBox is a destination of the frame
        - Enqueue to the link layer interface of the RedBox (with the PRP RCT)
      - If the frame is to be injected into the ring (RedBox is not sole destination and multicast/VLAN is ok)
        - If this is not the first occurrence of the frame at each HSR port
          - Register the occurrence
          - Discard the frame (already sent over that port)
        - Else (If this is the first occurrence of the frame at each HSR port)
          - Reuse the PRP-1 sequence number and the RedBox’s path identity to build the HSR tag
          - Enqueue the frame into each HSR port

Else (if the frame carries neither a HSR tag nor a PRP RCT)			
	If the source MAC address is not already registered in the ProxyNodeTable:		
		Create an entry in the proxy node table with a sequence number of 0;	
		Register that source as SAN	
	Else (If the source is already registered)		
		Register the presence of that source;	
	If the RedBox is a destination of the frame:		
		Enqueue to the link layer interface of the RedBox	
	If the frame is to be injected into the ring (RedBox is not sole destination and multicast/VLAN is ok))		
		If this is not the first occurrence of the frame at each HSR port	
			Register the occurrence
			Discard the frame (already sent over that port)
		Else (If this is the first occurrence of the frame at each HSR port)	
			Append the HSR tag using the sequence number of that node
			Increment the sequence number of that source;
			Enqueue the tagged frame into each HSR port

NOTE Reception of an HSR frame over the interlink is considered as a configuration error in PRP connection mode. If the RedBox is used as one half of a QuadBox in HSR connection mode, then the interlink will only carry HSR traffic. In HSR-HSR mode, the interlink is handled as an additional HSR port intended for QuadBox coupling and thus does not use a Proxy Node Table, unlike PRP-HSR and SAN-HSR RedBoxes. The RedBox implements a node table to record the presence of HSR nodes.

5.4.3 RedBox forwarding on the ring

In addition to the forwarding rules of 5.3.4, a RedBox shall not forward in the ring a unicast frame that is intended for one of the nodes that are registered in the proxy node table. This condition is enabled by default and can be disabled by setting the RedBox to Mode U.

5.4.4 RedBox receiving from an HSR port

A RedBox that receives a valid frame over one HSR port shall:

If this frame is not HSR-tagged:		
	If the frame is a link local traffic (IEEE 802.1D Table 7-10)	
		Consume the frame and do not forward it.
	Else if the node is in Mode M	
		Enqueue the unchanged frame for passing to the interlink.
		Forward according to 802.1D forwarding rules
	Else if the node is in Mode T	
		Set the port to non-HSR mode
		Enqueue the unchanged frame for passing to its link layer interface.
		Forward according to 802.1D forwarding rules
	Else	
		Enqueue the unchanged frame for passing to its link layer interface.
		Do not forward this frame
Else (frame is HSR-tagged):		
	If this is the first occurrence of the frame in direction of the second HSR port	
		Register the occurrence of the frame;
		Enqueue the frame to the second port;

	Else (If this is not the first occurrence of the frame in direction of the second HSR port)			
		Register the occurrence of the frame; Do not enqueue the unchanged frame to the second HSR port;		
	If this is the first occurrence of the frame in direction of the interlink:			
		Register the occurrence of the frame; If the RedBox is in SAN mode:		
			Remove the HSR tag;	
		Else if the RedBox is in PRP mode:		
			If the path Id matches that of the RedBox	
				Discard the frame
			Else	
			Remove the HSR tag and append the LanId (“A” or “B”) of the RedBox and reusing the HSR sequence number.	
		Else (if the RedBox is in HSR mode)		
			Do not modify the frame.	
		Enqueue frame for passing to the interlink.		
	Else (If this is not the first occurrence of the frame in direction of the interlink):			
		Register the occurrence; Discard the frame;		

NOTE A RedBox does not check if the frame was sent by one of the nodes for which it is a proxy since it cannot distinguish if the frame could have been sent by a redundant RedBox.

5.4.5 Redbox proxy node table handling

A RedBox shall hold a proxy node table containing an entry for each represented association, which shall support at least ProxyTableMaxEntries entries.

A RedBox shall purge the entry of a node in the Proxy Node Table with a configurable time-out (default of ProxyTableForgetTime) for non-receiving frames of this node.

NOTE The actual size of the proxy node table is indicated in the PICS.

5.4.6 RedBox CoS

Same as 5.3.5, except that RedBox is expected to support more VLANs and more complex and comprehensive options for engineering traffic flow in the network on higher protocol layers, e.g. multicast filtering rules, than a simple node.

Change the title of 5.4.7.

5.4.7 RedBox clock synchonization (informative)

Same as 5.3.6.

5.4.8 RedBox medium access

Same as 5.3.7, except that the RedBox must be able to aggregate several nodes for which it acts as a proxy.

5.5 QuadBox specification

A Quadbox shall operate conceptually as two RedBoxes in HSR-HSR mode, back-to-back.

5.6 Duplicate identification and discard

For the purpose of duplicate discard, a frame shall be identified by:

- its source MAC address;
- its sequence number.

NOTE It is possible to use other fields of the frame such as the checksum to aid in duplicate detection.

The duplicate discard method is not specified.

Any duplicate discard method shall be able to forget an entry identified by <Source MAC Address><Sequence number> after a time EntryForgetTime

5.7 Frame format for HSR

5.7.1 HSR-tagged frame format

5.7.1.1 Frame format for all frames

Frames to be treated HSR are identified uniquely by their source MAC address, destination MAC address and HSR tag.

If the frame carries a VLAN tag according to IEEE 802.1Q, it shall be inserted before the HSR tag.

The HSR tag is announced by the dedicated HSR_EtherType = 0xXXXX.

Editorial Note: An EtherType for HSR has applied for at IEEE EtherType Field Registration Authority and will replaced 0xXXXX during CDV.

a) 4-bit path identifier

b) 12 bit frame size (LSDU_size) or version if smaller than 64

c) 16-bit sequence number (sequenceNr)

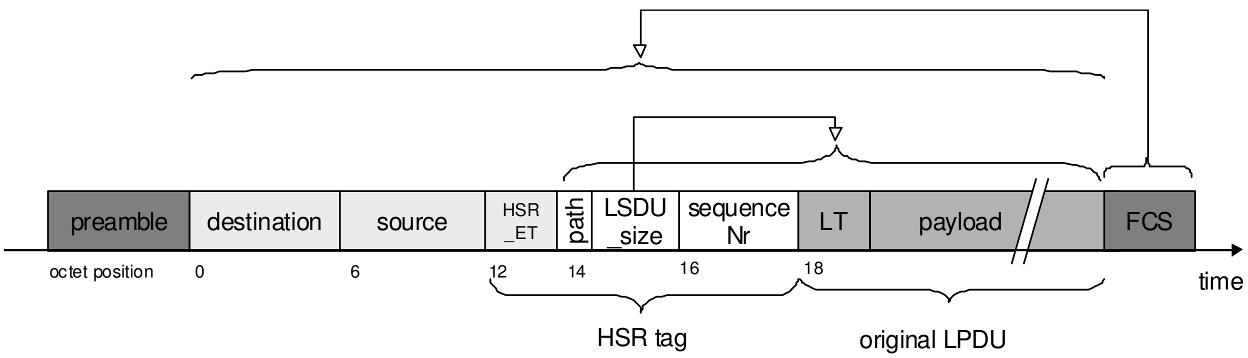

Figure 23 – HSR frame without VLAN tag

The definition of the 4-bit path field is:

- 0000 PRP management (supervision frames),
- 0001 ring identifier (regular HSR frames),
- 0010 – 1100: frames from PRP network
- 1111 HSR management (supervision frames)
- other: reserved.

The same frame with a VLAN tag is shown in Figure 24.

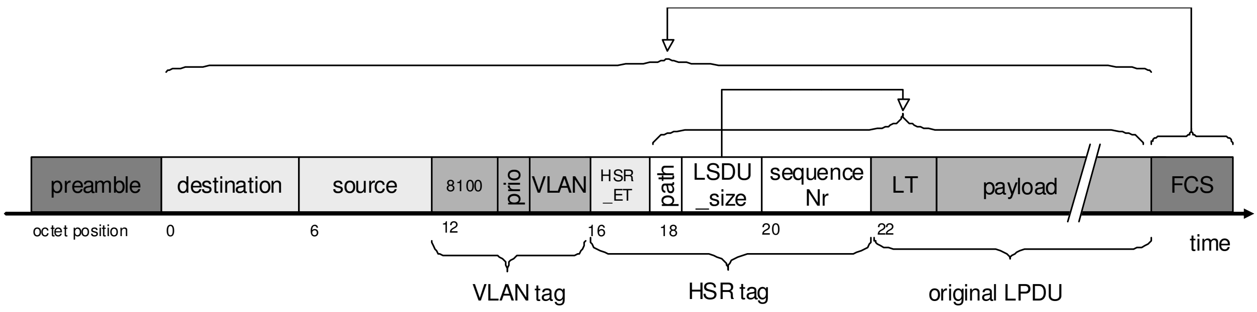

Figure 24 – HSR frame with VLAN tag

NOTE 1 The frames used within the ring include the HSR tag at the beginning of the frame to allow early identification of frames for cut-through operation.

NOTE 2 The LSDU_Size has the same definition as in PRP; it does not include the MAC header and VLAN tag, but includes the HSR tag itself and the original LPDU. The VLAN tag can be removed by intermediate switches.

NOTE 3 The reason for inserting the HSR tag after the VLAN tag is to provide faster MAC address lookup in case IVL (Independent VLAN learning as defined in IEEE 802.1Q) is used.

5.7.1.2 HSR_Supervision frame

5.7.1.2.1 Sending

Each DANH shall multicast a HSR_Supervision frame over both its ports with the format specified in Table 7 every LifeCheckInterval (see Table 8).

Table 7 – HSR_Supervision frame with optional VLAN tag

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0						msb	U/L	I/G								
2	HSR_destinationAddress = multicast (01-15-4E-00-01-XX)															
4									lsb							
6						msb	U/L	0								
8	HSR_sourceAddress (MAC address of the adapter)															
10									lsb							
12	VLAN Tag (=0x8100 for VLAN)															
14	prio			cti	vlan_identifier											
16	HSR_EtherType															
18	HSR_path				HSR_lsduSize											
20	HSR_sequenceNumber															
22	SupEthertype															
24	SupPath				SupVersion (< 64)											
26	SupSequenceNumber															
28	TLV.Type = 22								TLV.Length = 6							
30						Msb	U/L	0								
32	MacAddress (MAC address of the DANH)															
34																
36	TLV2.Type = 30								TLV2.Length = 6							
38						Msb	U/L	0								
40	RedBoxMacAddress															
42									lsb							
44	TLV3.Type = 31								TLV3.Length = 6							

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
46						msb	U/L	0								
48	VdanMacAddress															
50									lsb							
	Padding to 64 octets (no VLAN) or to 68 octets (VLAN)															
64	FCS															
66																

NOTE 1 The format of the frame is nearly identical to that of the PRP supervision frames. The use of two HSR Ethertype fields allows the hardware to remove the HSR tag without consideration of the frames contents before passing the frame to the link management layer, which is usually implemented in software.

NOTE 2 The NodesTable keeps track of the presence of other nodes identified by their MAC address. When a node does not send any frame within a given timeout (in the order of one minute), its address is removed from the NodesTable if such are used. .

5.7.1.2.2 HSR_Supervision frame contents

HSR_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.

HSR_sourceAddress
MAC address of the sending adapter.

VLAN tag (optional)
802.1Q tag if used. .

HSR_EtherType
Ethertype of the HSR protocol, see 5.7.1

HSR_Path
Path over which this frame was sent, in particular, identification of the PRP network.

HSR_LsduSize
Size of the LSDU.

HSR_SequenceNumber
SendSeq of the sender.

SupEthertype
Ethertype for supervision frames, shall be 0x88FB

SupLan
Path A or B over which the frame was sent

SupVersion
Indicates the protocol version, set to “1” (one) for this version of HSR.
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 exactly as specified for the version concerned. The version shall not exceed the value of 64.

SupSequenceNumber
Sequence number of the supervision frames

TLV.Type
Indicates the operation mode and shall have a value of 23 to indicate that it is an HSR node in normal operation. Other values are reserved.

TLV.Length
Indicates the length of the following MAC address.

MacAddress
MAC address used by the node. the sender.

TLV2
This field shall be set to 30 if the source node is a RedBox (see 4.2.7.6.3)

TLV2.Length
Indicates the length of the following MAC address.

RedBoxMacAddress
MAC addresses used by the RedBox on its interlink.

TLV3
This field shall be set to 31 for a VDANTLV3.Length
Indicates the length of the following MAC address.

VdanMacAddress
MAC addresses of the VDAN.

The VDAN fields can be repeated up to the maximum frame size. If this is not sufficient, a RedBox can send subsequent supervisory frames until all VDAN have been announced. A TLV4 = 0 shall close the list of VdanMacAddresses, a TLV4 = 1 shall indicate that the list is continued in a next supervisory frame.

NOTE 1 Octets with offset 14 to 17 are inserted only if a VLAN tag according to IEEE 802.1Q is used.

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

5.7.1.2.3 Reception of a HSR_Supervision frame

When receiving a first HSR_Supervision frame over any ring port, 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 of the frame, and register the port over which this frame came.

Subsequent reception of frames shall be registered with the objective of identifying reception errors.

5.7.1.2.4 Non-Reception of a HSR_Supervision frame

When a device detects another device as non-existent, it shall, if this device resumes transmission, accept frames from that device as if that device had not been connected before.

NOTE The mechanism to detect non-existing devices is not specified, it can be assumed that reading the NodesTable resets or reads a frame counter.

5.7.1.3 Constants

The constant parameters are shown in Table 8.

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

Table 8 – HSR Constants

Constant	Description	Default value
LifeCheckInterval	How often a node sends a HSR_Supervision frame	2 000 ms
NodeForgetTime	Time after which a node entry is cleared in the NodesTable	60 000 ms
ProxyTableForgetTime	Time after which a node entry is cleared in the ProxyTable	60 000 ms
ProxyTableMaxEntries	Maximum number of entries in the ProxyTable	512
EntryForgetTimel	Time after which an entry must be removed from the duplicates	400 ms
NodeRebootInterval	Minimum time during which a node that reboots remains silent	500 ms

MutlcastFilterSize	Number of multicast addresses to be filtered	16

6 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 reduced RSTP switch without designated port role
- PRP_RSTP: ability to perform as a RSTP switch element with designated port role
- PRP_MRP: ability to perform as a MRP switch element (client or master)
- HSR_H ability to perform as a HSR switch element
- HSR_N ability to use the HSR tag to reject duplicates, but without forwarding.
- HSR_M: ability to support mixed traffic (future use)
- HSR_RBX:RedBox capable of supporting singly attached nodes
- HSR_QBX:QuadBox integrating two RedBoxes
- HSR_PNT: number of entries in the proxy node table

7 PRP/HSR Management Information Base (MIB)

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 or HSR_MIB is true, the MIB data structures defined in this clause shall be available at OID = 1.0.62439 in addition to the MIBs that the adapters provide, with the following definition.

Editorial Note: This MIB should be validated. If errors are present, they will be corrected in the CDV

```
-- ***************************************************************************
IEC-62439-3-MIB DEFINITIONS ::= BEGIN

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

IMPORTS
   MODULE-IDENTITY,OBJECT-TYPE, Counter32,TimeTicks,
   Integer32, Unsigned32 FROM SNMPv2-SMI
   TruthValue,MacAddress   FROM SNMPv2-TC;


-- ***************************************************************************
-- Root OID
-- ***************************************************************************

iec62439 MODULE-IDENTITY

   LAST-UPDATED "201011090000Z" - November 9, 2010
   ORGANIZATION "IEC/SC 65C"
   CONTACT-INFO ""

   DESCRIPTION "This MIB module defines the Network Management interfaces
       for the redundancy protocols defined by the IEC 62439 suite."

   REVISION    "201010150000Z"  --  October 15, 2010
   DESCRIPTION  "
       Addendum to the IEC 62439-3 with changed to PRP and HSR. Changes in the
       MIB reflect the changes in the protocol definition."


   REVISION    "200811100000Z"  --  November 10, 2008
   DESCRIPTION  "
       Separation of IEC 62439 into a suite of documents
```

```
       This MIB applies to IEC 62439-3, added HSR functionality
                "

   REVISION  "200612160000Z" -- December 16, 2006
   DESCRIPTION "Initial version of the Network Management interface for the
       Parallel Redundancy Protocol"

   ::= {iso std(0) 62439 }

-- ***********************************************************************
-- Redundancy Protocols
-- ***********************************************************************
mrp                 OBJECT IDENTIFIER ::= { iec62439 2 }
prp                 OBJECT IDENTIFIER ::= { iec62439 3 }
crp                 OBJECT IDENTIFIER ::= { iec62439 4 }
brp                 OBJECT IDENTIFIER ::= { iec62439 5 }
drp                 OBJECT IDENTIFIER ::= { iec62439 6 }
rrp                 OBJECT IDENTIFIER ::= { iec62439 7 }
-- ***********************************************************************
-- Objects of the PRP Network Management
-- ***********************************************************************

manufacturerName OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(1..255))
  MAX-ACCESS read-only
  STATUS  current
   DESCRIPTION
    "specifies the name of the device manufacturer"
   ::= { prp 1 }

interfaceCount  OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
         "The total number of LRE interfaces present in this system."
::= { prp 2 }

redundantInterfaces OBJECT IDENTIFIER ::= { prp 3}

redundantInterfaceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RedundantInterfaceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A list of PRP/HSR redundant interfaces. Each entry corresponds
            to one redundant PRP/HSR Link Redundancy Entity (LRE) with a
            LAN connector A and B each. A basic device supporting PRP/HSR might
            have only one LRE and thus one entry in the table, while more complex
            devices might have several entries for multiple LRE's."
    ::= { redundantInterfaces 1 }

redundantInterfaceEntry OBJECT-TYPE
    SYNTAX      RedundantInterfaceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry containing management information applicable to a
            particular LRE."
    INDEX   { redundantInterfaceIndex }
    ::= { redundantInterfaceTable 1 }

RedundantInterfaceEntry ::=
   SEQUENCE {
   redundantInterfaceIndex    Unsigned32,
   nodeType                   INTEGER,
   nodeName                   OCTET STRING,
   versionName                OCTET STRING,
   macAddressA                MacAddress,
   macAddressB                MacAddress,
   adapterActiveA             INTEGER,
   adapterActiveB             INTEGER,
   linkStatusA                INTEGER,
   linkStatusB                INTEGER,
   duplicateDiscard           INTEGER,
   transparentReception       INTEGER,
   hsrLREMode                 INTEGER,
```

```
   switchingEndNode           INTEGER,
   cntTotalSentA              Counter32,
   cntTotalSentB              Counter32,
   cntErrWrongLANA            Counter32,
   cntErrWrongLANB            Counter32,
   cntReceivedA               Counter32,
   cntReceivedB               Counter32,
   cntErrorsA                 Counter32,
   cntErrorsB                 Counter32,
   timeLastSeenA              TimeTicks,
   timeLastSeenB              TimeTicks,
   sanA                       TruthValue,
   sanB                       TruthValue,
   cntNodes                   INTEGER,
   evaluateSupervision        TruthValue,
   nodesTableClear            INTEGER
}

redundantInterfaceIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION "A unique value for each LRE interface."
   ::= { redundantInterfaceEntry 1 }

nodeType OBJECT-TYPE
   SYNTAX INTEGER{
    prp-mode-0 (1),
    prp-mode-1 (2),
    hsr (3)
    }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
    "specifies the operation mode of the redundant interface:
    PRP Mode 0 (1)
    PRP Mode 1 (2)
    HSR        (3)"
   ::= { redundantInterfaceEntry 2 }

nodeName OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(1..32))
   MAX-ACCESS read-write
   STATUS  current
   DESCRIPTION
    "specifies this LRE's node name"
   ::= { redundantInterfaceEntry 3 }

versionName OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(1..32))
   MAX-ACCESS read-only
   STATUS  current
   DESCRIPTION
    "specifies the version of this LRE's software"
   ::= { redundantInterfaceEntry 4 }

macAddressA OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-write
   STATUS  current
   DESCRIPTION
    "Specifies the MAC address to be used by network interface A of this LRE"
   ::= { redundantInterfaceEntry 5 }

macAddressB OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-write
   STATUS  current
   DESCRIPTION
    "Specifies the MAC address to be used by network interface B of This LRE"
   ::= { redundantInterfaceEntry 6 }

adapterActiveA OBJECT-TYPE
   SYNTAX INTEGER {
    notActive (1),
    active (2)
    }
   MAX-ACCESS read-write
```

```
   STATUS  current
   DESCRIPTION
    "Specifies whether the adapter A shall be active or not Active through
    administrative action"
   ::= { redundantInterfaceEntry 7 }

adapterActiveB OBJECT-TYPE
   SYNTAX INTEGER {
    notActive (1),
    active (2)
    }
   MAX-ACCESS read-write
   STATUS  current
   DESCRIPTION
    "Specifies whether the adapter B shall be active or not Active through
    administrative action"
   ::= { redundantInterfaceEntry 8 }

linkStatusA OBJECT-TYPE
   SYNTAX INTEGER {
    down (1),
    up (2)
    }
   MAX-ACCESS read-only
   STATUS  current
   DESCRIPTION
    "shows the actual link status of the LRE's port A"
   ::= { redundantInterfaceEntry 9 }

linkStatusB OBJECT-TYPE
   SYNTAX INTEGER {
    down (1),
    up (2)
    }
   MAX-ACCESS read-only
   STATUS  current
   DESCRIPTION
    "shows the actual link status of the LRE's port B"
   ::= { redundantInterfaceEntry 10 }

duplicateDiscard OBJECT-TYPE
   SYNTAX INTEGER {
    doNotDiscard (1),
    discard (2)
    }
   MAX-ACCESS read-write
   STATUS  current
   DESCRIPTION
    "specifies whether a duplicate discard algorithm is used at reception and,
    if the interface is a PRP interface, that the RCT is appended at sending"
   ::= { redundantInterfaceEntry 11 }

transparentReception OBJECT-TYPE
   SYNTAX INTEGER {
    removeRCT (1),
    passRCT (2)
    }
   MAX-ACCESS read-write
   STATUS  current
   DESCRIPTION
    "if 1, the RCT is removed when forwarding to the upper layers, only applicable
    for PRP interface."
   ::= { redundantInterfaceEntry 12 }

hsrLREMode OBJECT-TYPE
   SYNTAX INTEGER {
   mode-h (1),
   mode-n (2),
   mode-t (3),
   mode-u (4)
   mode-m (5)
   }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
   "This enumeration is only applicable if the LRE is an HSR switching node or RedBox.
    It shows the mode of the HSR LRE:
    (1) Default mode: The HSR LRE is in mode h and switches tagged HSR traffic
```

```
    (2) Optional mode: The HSR LRE is in mode n and has switching between its HSR ports
disabled.
      Traffic is HSR tagged.
    (3) Optional mode: The HSR LRE is in mode t and switches non-tagged HSR traffic
between its
      HSR ports
    (4) Optional mode: The HSR LRE is in mode u and behaves like in mode h, except it
does not remove
        unicast messages"
   ::= { redundantInterfaceEntry 13}

switchingEndNode OBJECT-TYPE
   SYNTAX INTEGER {
    prp-node(1),
    hsr-node(2),
    hsr-redboxsan(3),
    hsr-redboxhsr(4),
    hsr-redboxprpa(5),
    hsr-redboxprpb(6)

    }
   MAX-ACCESS read-write
   STATUS  current
   DESCRIPTION
    "This enumeration shows which feature is enabled in this particular LRE:
    (1): a PRP node/RedBox.
    (2): an HSR switching node.
    (3): an HSR RedBox with regular Ethernet traffic on its interlink.
    (4): an HSR RedBox with HSR tagged traffic on its interlink.
    (5): an HSR RedBox with PRP traffic for LAN A on its interlink.
    (6): an HSR RedBox with PRP traffic for LAN B on its interlink.
"
   ::= { redundantInterfaceEntry 14 }

cntTotalSentA OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS  current
   DESCRIPTION
    "number of frames sent over network interface A"
   ::= { redundantInterfaceEntry 15 }

cntTotalSentB OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS  current
   DESCRIPTION
    "number of frames sent over network interface B"
   ::= { redundantInterfaceEntry 16 }

cntErrWrongLANA OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "number of frames with the wrong LAN identifier received on interface A"
   ::= { redundantInterfaceEntry 17}

cntErrWrongLANB OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "number of frames with the wrong LAN identifier received on LRE's interface B"
   ::= { redundantInterfaceEntry 18}

cntReceivedA OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "number of frames received on a LRE's interface A"
   ::= { redundantInterfaceEntry 19}

cntReceivedB OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
```

```
   DESCRIPTION
   "number of frames received on a LRE's interface B"
   ::= { redundantInterfaceEntry 20}

cntErrorsA OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS  current
   DESCRIPTION
    "number of frames with errors received from this LRE's interface A"
   ::= { redundantInterfaceEntry 21}

cntErrorsB OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS  current
   DESCRIPTION
    "number of frames with errors received from this LRE's interface B"
   ::= { redundantInterfaceEntry 22 }

timeLastSeenA OBJECT-TYPE
   SYNTAX TimeTicks
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "UTC time at which the latest frame was received over LAN_A"
   ::= { redundantInterfaceEntry 23 }

timeLastSeenB OBJECT-TYPE
   SYNTAX TimeTicks
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "UTC time at which the latest frame was received over LAN_B"
   ::= { redundantInterfaceEntry 24 }

sanA OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "True if the remote device is most probably a SAN accessible over adapter A"
   ::= { redundantInterfaceEntry 25}

sanB OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
   "True if the remote device is most probably a SAN accessible over adapter B"
   ::= { redundantInterfaceEntry 26}

cntNodes OBJECT-TYPE
   SYNTAX INTEGER
   MAX-ACCESS read-only
   STATUS  current
   DESCRIPTION
    "number of nodes in the Nodes Table"
   ::= { redundantInterfaceEntry 27}

evaluateSupervision OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
   "True if the LRE shall evaluate received supervision frames. False if it
   shall drop the supervision frames without evaluating. Note: LREs are required
   to send supervision frames, but reception is optional."
  ::= { redundantInterfaceEntry 28}

nodesTableClear OBJECT-TYPE
   SYNTAX INTEGER {
    noOp (0),
    clearNodesTable (1)
    }
   MAX-ACCESS read-write
   STATUS  current
   DESCRIPTION
```

```
    "specifies that the Nodes Table is to be cleared"
   ::= { redundantInterfaceEntry 29}

END
```

Annex A
(informative)
PRP duplicate discard algorithm as pseudo-code

A.1 Constants

```
integer32     MaxErrors;           // maximum number of errors considered
timeMilli     LifeCheckInterval;   // how often the presence of a node is checked
timeMilli     NodeForgetTime;      // time after which node entry is cleaned
integer16     DropWindowMax;       // max size of capture window
integer16     TwoPi                // window size = DropWindowMax
integer16     OnePi                // half the window size = DropWindowMax / 2
integer16     NodeTableEntryNrMax  // max number of entries in the NodeTable
```

A.2 Data structures

A.2.1 Base data types

```
integerXX                          // integer with a size of XX bits
octetString                        // string of unspecified octets
timeMicro                          // time in microseconds (32 bits)
timeMilli                          // time in milliseconds (32 bits)
boolean1                           // boolean that is not part of a set
```

A.2.2 Ethernet frame

This structured data type expresses a frame processed by the driver:

```
typedef FrameType = struct {
 integer48    sourceMacAddress;
 integer12    r_size;              // field before CRC
 integer4     r_LAN;               // nibble in length filled before CRC
 integer16    r_SequenceNr;        // sequence number before CRC
 integer16    physicalSize;        // size as detected by the controller
 timeMilli    timeStamp;           // time of reception
 octetString  lsdu                 // payload, not used in algortihm
 }
```

A.2.3 Source device

This structured data type expresses each source device in the LRE:

```
typedef SourceType = struct {
 integer48      nodeMacAddress;        // normally identical to nodeMACAddressA
 integer48      nodeMacAddressB;       // in case they are different
 timeMilli      timNodeLastSeen;
 integer16      cntStartSeqLanA;       // sequence number that starts the interval
 integer16      cntStartSeqLanB;
 integer16      cntExpectedtSeqLanA;   // next expected sequence number
 integer16      cntExpectedtSeqLanB;
 timeMilli      lastTimeReceivedLanA;  // time of latest reception
 timeMilli      lastTimeReceivedLanB;  // time of latest reception
 integer32      cntErrWrongLanA;       // error counter
 integer32      cntErrWrongLanB;
 integer32      cntErrOutOfSequenceA;    // error counter
 integer32      cntErrOutOfSequenceB;
 enum           stateLanA;             // normal, disabled
 enum           stateLanB;             // normal, disabled
 }
```

A.2.4 Receiver

This structured data type expresses the receiver state:

```
 typedef ReceiverType = struct {
  integer16     sourceQty;             // quantity of registered sources
  integer32     cntErrorsLanA;         // sum of errors on LAN_A
  integer32     cntErrorsLanB;         // sum of errors on LAN_B
  SourceType    sources[0..NodeTableEntryNrMax];      // number of expected partners
  }

 typedef senderType = struct {
  sendSequenceNr;                      // valid for both LANs
```

```
  }
```

A.3 Procedures

A.3.1 Sender initialization

```
 sendSequenceNr = 0;                     // but could be random as well
```

A.3.2 Sending a frame

```
 frame.r_size = computeFrameSize(frame);
 frame.sequenceNr = sendSequenceNr;
 sendSequenceNr = sendSequenceNr + 1;    //modulo TwoPi = 65536

 frame.r_LAN = 0xA; // the sequence number is the same on both LANs
 send(frame, LANA);
 frame.r_LAN = 0xB; //
 send(frame, LANB);
```

A.3.3 Receiver initialization

```
 SourceType   sourceList [0..MaxSourceNr-1];
 ReceiverType receiver;
 Initialize(receiver)
```

A.3.4 Receiver reception of a frame

```
// this modulo arithmetic is simplified to work with 16-bit registers.
// the modulo arithmetic is emulated with the TwoPi and OnePi constants

if (frame.r_size == frame.physicalSize) &&
 ((frame.r_LAN == LANA) || (frame.r_LAN == LANB)) {
                // frame with redundancy info
 if (~ InSourceList(frame.source)) {
  Insert (frame.source, sourceList, index);  // register only DANP sources
  Initialize_source_object (frame.source);
  }
                // known node
  // thisLAN = LAN over which frame was received (can be a field in frame)
  otherLAN = (thisLAN + 1) Mod 2;      // index of other LAN
  currentSeq(thisLAN) = sequenceNr;
  if (((currentSeq(thisLAN) - startSeq(otherLAN) + TwoPi) Mod TwoPi) =< OnePi) _
  && (((expectedSeq(otherLAN) - currentSeq(thisLAN) + TwoPi -1) Mod TwoPi) < OnePi) {
                 // drop frame
  if ~ (currentSeq(thisLAN) == expectedSeq(thisLAN)) {
                 // check sequence
   cntErrOutOfSequence(thisLAN) = (cntErrOutOfSequence(thisLAN) + 1)
                 // increase seq errors for A or B
  }

  expectedSeq(thisLAN) = (currentSeq(thisLAN) + 1) Mod TwoPi
                 // new expected sequence nr
  startSeq(otherLAN) = expectedSeq(thisLAN);
                 // reduce other window
  startSeq(thisLAN) = expectedSeq(thisLAN)
                 // disable this LAN
  Drop (thisLAN)
                 // drop, already received
  }
  else {
             // forward frame
   if (~ (currentSeq(thisLAN) == expectedSeq(thisLAN)) {
             // check monotonicity of sequence
    cntErrOutOfSequence(thisLAN) = (cntErrOutOfSequence(thisLAN) + 1)
             // increase sequence errors
    startSeq(thisLAN) = currentSeq(thisLAN)
             // reset dropWindow to one
    }
   else {
             // correct sequence, slide window
    if ((expectedSeq(thisLAN) - startSeq(thisLAN) + TwoPi) Mod TwoPi >
    dropWindowMax) {
     if expectedSeq(otherLAN) == startSeq(thisLAN) {
               // register sequence error
      cntErrStall(otherLAN) = cntErrStall(otherLAN) + 1
      }
     startSeq(thisLAN) = (expectedSeq(thisLAN) + TwoPi - dropWindowMax)
     Mod TwoPi
               // adjust window
```

```
    }            // slide window
   }             // correct sequence
  startSeq(otherLAN) = expectedSeq(otherLAN)
               // disable the other LAN
  expectedSeq(thisLAN) = (currentSeq(thisLAN) + 1) Mod TwoPi
               // new expected sequence nr
  Forward_To_UpperLayer (thisLAN)
 }
```

A.3.5 Timeout process

```
// execute at CheckLiveTime interval

for each source in sourcelist do
 if (source.timeLastSeenA - currentTime) then
  source.missingErrorLANB++// just register the error, no impact on algorithm
 endif
```