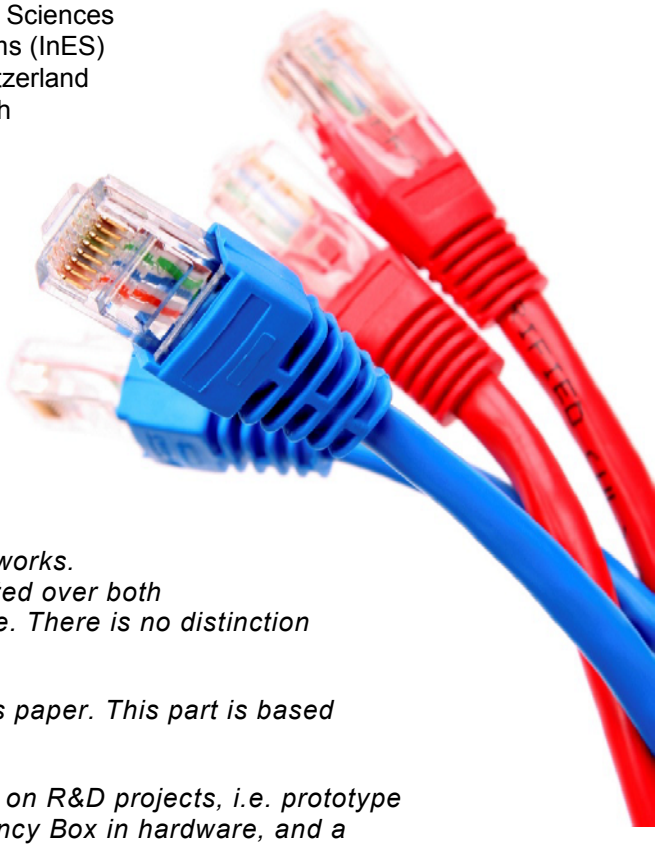
# Tutorial on Parallel Redundancy Protocol (PRP)

Prof. Hans Weibel
Zurich University of Applied Sciences
Institute of Embedded Systems (InES)
CH-8401 Winterthur, Switzerland
hans.weibel@zhaw.ch



## Abstract

*High availability applications typically count on the network's ability to reconfigure in case of a failure. The Parallel Redundancy Protocol (PRP) follows a different approach from the well-known reconfiguration protocols. It makes use of two independent Ethernet networks. Frames are replicated by the sending node and transmitted over both networks. Duplicates are discarded by the receiving node. There is no distinction between a working and a backup path.*

*The basic operation principles of PRP are outlined in this paper. This part is based on the standards document.*

*Implementation issues are based on experiences gained on R&D projects, i.e. prototype implementations of an end node for Linux, of a Redundancy Box in hardware, and a redundancy manager application.*

## Table of Contents

1 Introduction to high-availability Ethernet protocols 2
2 PRP principle of operations 3
2.1 Network and network elements 3
2.2 Structure of a DAN 4
2.3 Communication between SANs and DANs 4
3 Methods for handling duplicates 5
3.1 Duplicate accept 5
3.2 Duplicate discard in the link layer 5
3.3 Duplicate discard algorithm 7
3.4 Nodes table 9
4 Network supervision 10
5 Redundancy management interface 12
6 Configuration rules and parameters 12
7 Implementing PRP 13
7.1 Kernel mode variant of PRP software stack 13
7.2 User mode variant of PRP software stack 14
7.3 FPGA-based implementation of a RedBox 14
7.4 PRP redundancy manager 15
7.5 PRP combined with IEEE 1588 synchronization 19
Abbreviated terms and acronyms 20
References 20

# 1 Introduction to high-availability Ethernet protocols

Industrial real-time Ethernets typically demand for much higher availability and uninterrupted operation than office Ethernet solutions can provide. Even a short loss of connectivity may result in loss of functionality, as for example in some automation, vehicular, power generation, and power distribution systems.

To recover from a network failure, different standard redundancy schemes are proposed and applied, such as the Rapid Spanning Tree Protocol (RSTP) [3], Media Redundancy Protocol (MRP) [2], Parallel Redundancy Protocol (PRP) [1], and others.

RSTP can be applied in arbitrary mesh topologies. It implements a distributed computation of a tree based on path costs and priorities. This tree is the active topology which is established by blocking bridge ports. In case of failure, the tree is reconfigured, typically within a few seconds time range.

MRP is restricted to ring topology. A dedicated node, the ring manager, blocks one of its ring ports in order to establish a line as the active topology. In case of failure, this line breaks into two isolated lines which are reconnected by de-blocking the previously blocked port. Reconfiguration time is in the 100 ms range and can be guaranteed.

PRP does, in contrast to the methods described above, not change the active topology. It operates on two independent networks. Each frame is replicated on the sending node and transmitted over both networks. The receiving node processes the frame arriving first and discards the subsequent copy. The PRP layer is responsible for this replicate/discard function and hides the two networks from the upper layers. This scheme works without explicit reconfiguration and switchover and therefore does not show a period of unavailability.

The choice of a redundancy protocol depends on the application's tolerance against short interruptions caused by the switchover. The grace time of a high availability application is defined as the time the application allows for recovery before taking emergency actions (e.g. emergency shut-down, fall-back mode). Recovery time after a communication failure must be shorter than the grace time to pass unnoticed by the application. Critical applications such as synchronized drives, robot control, power substations, X-by-wire, and others may require a grace time in the order of 10 ms or less.

An RSTP or MRP reconfiguration includes failure detection, distributed execution of the reconfiguration algorithm, flushing and re-establishing (learning) the forwarding database (MAC address tables). The resulting switchover may be too slow for the most demanding applications. That's where PRP comes into play.

# 2 PRP principle of operations

## 2.1 Network and network elements

PRP implements redundancy functions in the end nodes rather than in network elements. This is one mayor difference to protocols like RSTP or MRP. An end node is attached to two similar LANs of arbitrary topology, which are disjoint and operated in parallel. The LANs must be laid out so that the two LANs fail independently e.g. redundant LANs must not be powered out of the same source. No direct connection can be made between the two LANs.

Figure1 depicts the general network as two switched networks, which can have any topology, e.g. tree, ring or meshed.

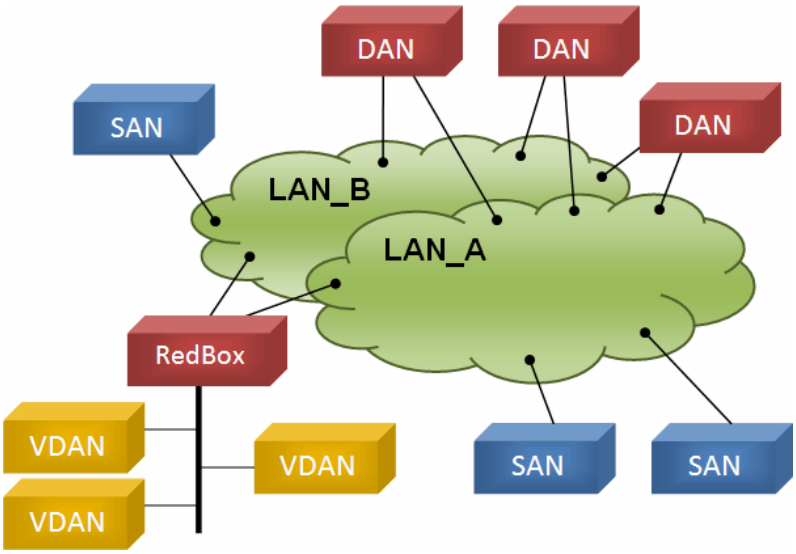



**Figure 1: PRP redundant network**

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

In some applications, only availability-critical nodes need a double attachment, while others do not. In order to meet the specific requirements, PRP defines different kinds of end nodes.

- The Dual Attached Node (DAN) is connected to both LANs.
- Uncritical nodes can be attached to only one LAN and are therefore called Single Attached Nodes (SAN). SANs that need to communicate with each other are on the same LAN.
- The Redundancy Box (RedBox) is used when a single interface node has to be attached to both networks. Such a node can communicate with all other nodes. Since a node behind a RedBox appears for other nodes like a DAN, it is called Virtual DANs (VDAN). The RedBox itself is a DAN and acts as a proxy on behalf of its VDANs. The RedBox has its own IP address for management purposes.

## 2.2 Structure of a DAN

Each DAN 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 2 shows.

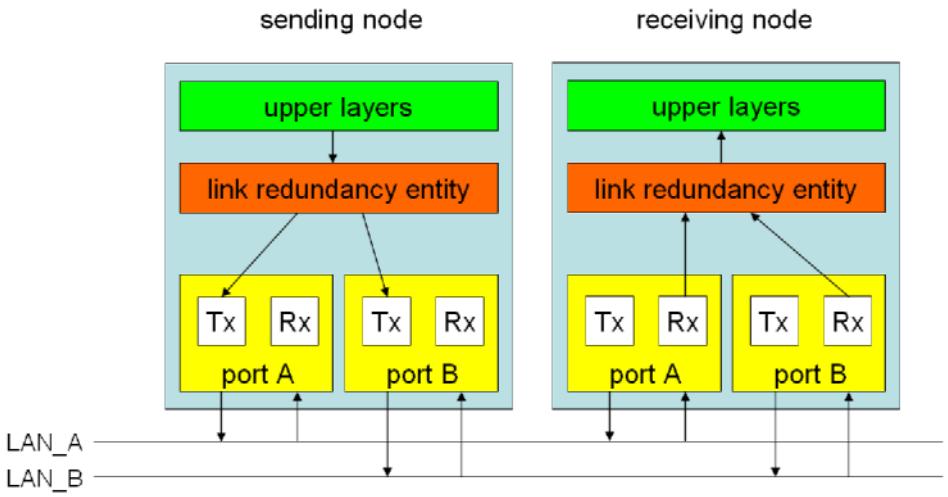



**Figure 2: Communication between two DANs**

When an upper layer protocol sends a frame, the LRE replicates the frame and sends it 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 at the destination node within a small time frame. When receiving, a node's LRE forwards the first received frame to its upper layers and discards the duplicate frame.

The LRE generates and handles duplicates. This layer presents toward its upper layers the same interface as the network adapter of a non-redundant adapter. The LRE has two tasks: handling of duplicates and management of redundancy. To supervise redundancy, the LRE appends to each sent frame a 32-bit redundancy control trailer (RCT) and removes that RCT at reception.

A DAN has the same MAC address on both ports, and only one set of IP addresses. This makes redundancy transparent to the upper layers. Especially, this allows the Address Resolution Protocol (ARP) to work the same as with a SAN.

## 2.3 Communication between SANs and DANs

SANs can be connected to any LAN. A SAN connected to one LAN can not communicate directly to a SAN connected to the other LAN. Since SANs do not implement any redundancy features, DANs have to generate frames that these SANs understand. The condition is however that the SANs ignore the RCT in the frames, which should be the case since a SAN cannot distinguish the RCT from IEEE 802.3 padding. Conversely, DANs understand the frames generated by SANs, since these do not append an RCT. They only forward just one frame to their upper layers since the SAN traffic uses one LAN only. If a DANP cannot identify that the remote node is a DAN, it inserts no RCT.

# 3 Methods for handling duplicates

Since a DAN receives the same frame over both adapters (if both are operational), it should keep one and ignore the duplicate. There are two methods for handling duplicates:

- Duplicate accept, in which the sender uses the original frame and the receiver forwards both frames it receives to its upper protocol layers and
- Duplicate discard, in which the sender appends a redundancy control trailer to both frames it sends and the receiver uses that redundancy control trailer to filter out duplicates.

## 3.1 Duplicate accept

This method does not attempt to discard duplicates at the link layer. The sender 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 handle duplicates (IEEE 802.1D explicitly states that higher layer protocols have to cope with duplicates).

The internet transport protocols UDP and TCP are assumed to be resilient against duplicates. The TCP protocol is designed to reject duplicates, so it will discard the second frame of a pair. The UDP layer is by definition connectionless and unacknowledged. All applications that use UDP must 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 such as ICMP and ARP are not affected by duplicates, since they have their own sequence numbering.

Therefore, one can assume that handling of duplicates is taken care of by the standard network protocols, but one has to check if each application complies with these assumptions and delivers the expected performance under these conditions.

This simple duplicate accept method has the disadvantage of not providing network supervision, since it does not keep track of correct reception of both frames.

## 3.2 Duplicate discard in the link layer

It is advantageous to discard duplicates already at the link layer for the purpose of offloading the application processor and for improving the redundancy supervision. The LRE can perform duplicate rejection, possibly with an independent pre-processor or an intelligent Ethernet controller.

The duplicate discard protocol uses an additional four octet wide field in the frame, the redundancy control trailer (RCT), which the LRE inserts into each frame that it receives from the upper layers, as figure 3 shows. The RCT consists of the following parameters:

- a 16-bit sequence number
- a 4-bit LAN identifier, 1010 (0xA) for LAN_A and 1011 (0xB) for LAN_B
- a 12 bit frame size

Appending the RCT could generate oversize frames that exceed the maximum frame size allowed by IEEE 802.3-2005. To maintain compliance with the IEEE 802.3-2005 standard, the communication software in a DAN using RCT shall be configured for a maximum payload size of 1496 octets.

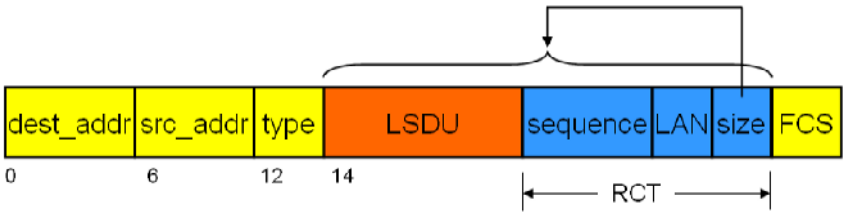



**Figure 3: Frame extended by an RCT**

Each time the link layer sends a frame to a particular destination the sender increases the sequence number corresponding to that destination and sends the (nearly) identical frames over both LANs.

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

To allow the receiver to distinguish easily frames coming from nodes that obey to the PRP from the non-redundant ones, the sender appends to the frame the length of the link service data unit (LSDU) in octets in the 12-bit frame size field. In VLANs, frame tags may be added or removed during transit through a switch. To make the length field independent of tagging, only the LSDU and the RCT are considered in the size.

The receiver scans the frames starting from the end. If it detects that the 12 bits before the end correspond to the frame size, and that the LAN identifier matches the identifier of the LAN it is attached to, 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 4 shows.

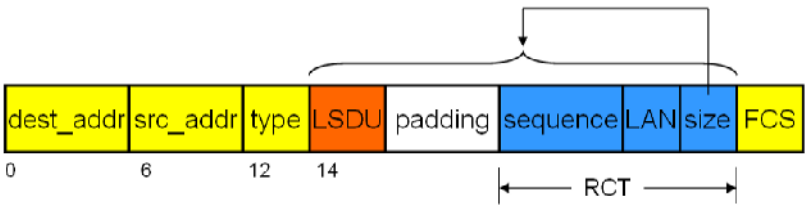



**Figure 4: Padded frame extended by an RCT**

A special situation has to be considered: A sender generates a VLAN tagged short frame padded to the size of 64 octets. This frame transits an intermediate node where the tag is removed and therefore a second padding after the RTC is applied, as depicted in figure 5. If the minimum size of a VLAN tagged frame is set to 68 octets instead of 64, this situation should never happen.

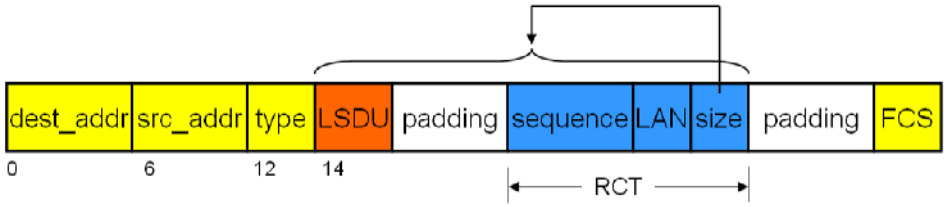



**Figure 5: Dual padded frame extended by an RCT as it can be seen by the receiver**

## 3.3 Duplicate discard algorithm

The receiver assumes that frames coming from each source that obeys to the PRP are sent in sequence with increasing sequence numbers. The sequence number expected for the next frame is kept in the variables ExpectedSeqA, resp. ExpectedSeqB.

At reception, the correct sequence can be checked by comparing ExpectedSeqA, resp. expectedSeq with the received sequence number in the RCT, currentSeq. Regardless of the result, expectedSeq is set to one more than currentSeq to allow checking the next expected sequence number on that line.

Both LANs thus maintain a sliding drop window of contiguous sequence numbers, the upper bound being expectedSeq (the next expected sequence number on that LAN), excluding that value, the lower bound being startSeq (the lowest sequence number that will lead to a discard on that LAN) as figure 6 shows for LAN_A.

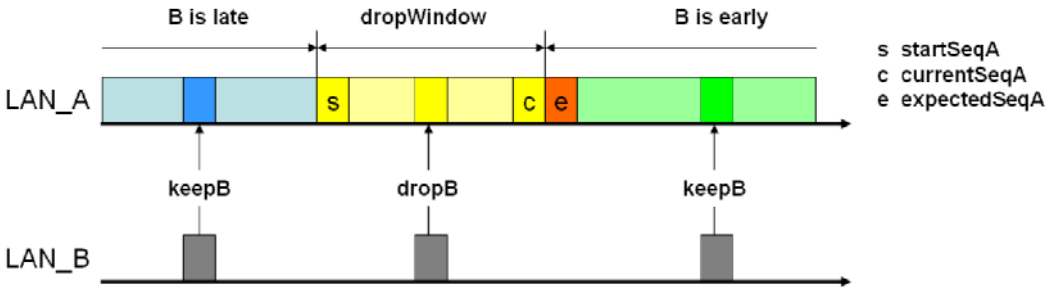



**Figure 6: Drop window (for LAN_A)**

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, a frame from LAN_B whose sequence number fits into the drop window of A will be discarded (dropB on figure 6). In all other cases, the frame is kept and forwarded to the upper protocol layers (keepB on figure 6).

Discarding the frame (dropB on figure 6) shrinks the drop window size on LAN_A since no more frames from LAN_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 LAN_B lags behind LAN_A and no frames from LAN_A should be discarded, as figure 7 shows.

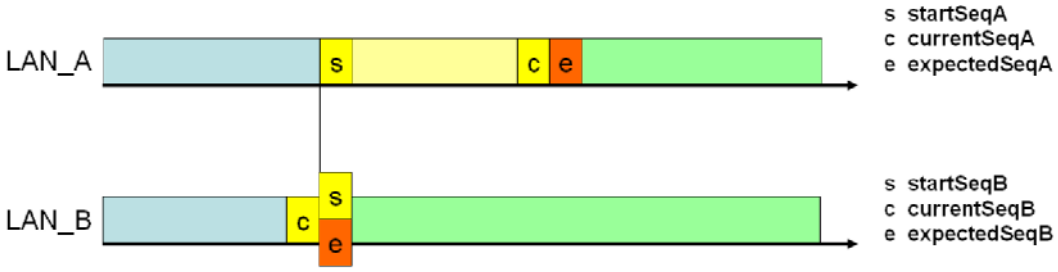



**Figure 7: Drop window reduction after a discard**

In the situation of figure Figure 8, if several frames come in sequence over the same LAN_A, but none on LAN_B, they will be kept since their currentSeq 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 will be 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 will be discarded, while the drop window of the other line is reset to 0, meaning that no frame will be discarded, as figure 8 shows.

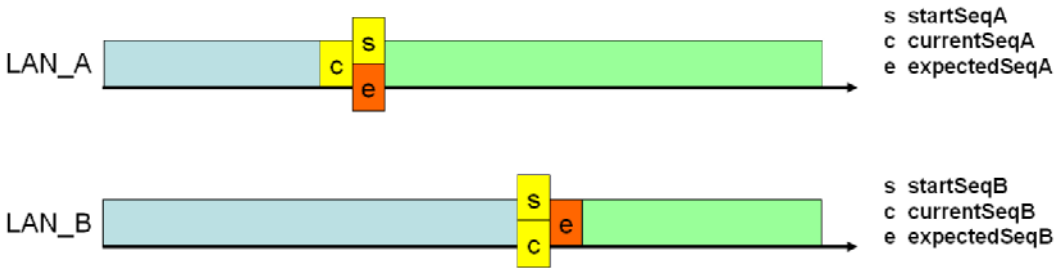



**Figure 8: Frame from LAN_B was not discarded**

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 will be kept and the drop window will be opened by one to allow only a frame with the same sequence number as the one already received, as figure 9 shows.

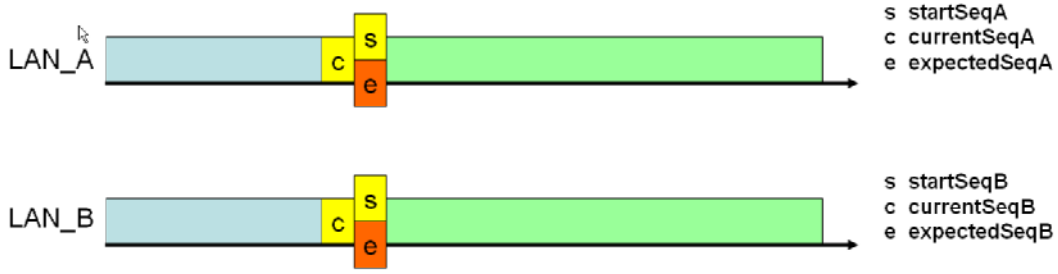



**Figure 9: Synchronized LANs**

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

There is no change to this algorithm when frames arrive out of sequence. This can happen if layer 2 prioritization is applied.

Because of the LAN identifier field in the RCT, the duplicate frames differ in one bit (and in the FCS). The receiver checks that the frame comes from the correct LAN. It will 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 will count an error cntWrongLanA or cntWrongLanB since this could hint at a configuration error. Since this kind of error is permanent, it will be detected rapidly.

## 3.4 Nodes table

A node shall maintain a table with an entry for each node (SAN or DAN) to which it sends or from which it receives frames. The nodes table's purpose is to support the discard algorithm and allow for network integrity monitoring. The table will contain one row for each unicast, multicast, and broadcast address this node is sending to. Table 1 lists the entries maintained per destination.

| Parameter | Description |
|---|---|
| sendSeq | a 16-bit sequence number used by this node for sending to that remote node or multicast or broadcast address (wrapping through zero) |
| expectedSeqA<br>expectedSeqB | for each adapter A and B, a 16-bit sequence number indicating the sequence number used last by the remote node to communicate with this node on that LAN, incremented by one (wrapping through zero) |
| cntErrOutOfSequenceA<br>cntErrOutOfSequenceB | for each adapter A and B, a 32-bit error counter indicating that a frame from the remote node was not received in sequence over that LAN |
| startSeqA<br>startSeqB | for each adapter A and B, a 16-bit cursor that limits the drop window |
| cntReceivedA<br>cntReceivedB | for each adapter A and B, a 32-bit counter indicating the number of frames received over the adapter |
| cntErrWrongLanA<br>cntErrWrongLanB | for each adapter A and B, a 32-bit counter indicating the number of mismatches on each adapter |
| timeLastSeenA<br>timeLastSeenB | for each adapter A and B, a time field indicating when this node received last a frame from the remote node |
| sanA<br>sanB | for each adapter A and B, a boolean indicating that the remote node is probably a SAN and/or that the remote node uses Duplicate Accept |

**Table 1: Nodes table entries**

## 4 Network supervision

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

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

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

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

Since the protocol is loosely connection oriented, the sequence numbers corresponding to non-existent nodes must be cleaned up.

Supervision relies on each DAN sending periodically a 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 DAN, the MAC addresses they use and which operating mode they support, duplicate accept or duplicate discard.

Figure 10 and 11 depict the untagged and the tagged supervision frame format and the individual fields of the supervision frames are explained in table 2.

| Offset | Length | 0 7 | 8 15 |
|---|---|---|---|
| 0 | 6 | PRP_DestinationAddress (multicast 01-15-4E-00-01-XX) | |
| 6 | 6 | PRP_SourceAddress (MAC address of the DAN) | |
| 12 | 2 | Type (0x88FB for PRP) | |
| 14 | 2 | PRP_Ver | |
| 16 | 2 | PRP_TLV.Type = 20 or 21 | PRP_TLV.Length = 12 |
| 18 | 6 | PRP_SourceMacAddressA (MAC address A of the DAN) | |
| 24 | 6 | PRP_SourceMacAddressB (MAC address B of the DAN) | |
| 30 | 2 | PRP_TLV.Type = 30 or 31 | PRP_TLV.Length = 6 |
| 32 | 6 | PRP_SourceMacAddressA (MAC address A of the RedBox or VDAN) | |
| 38 | 22 | padding to 64 octets | |
| 60 | 2 | SequenceNr | |
| 62 | 2 | LAN = A or B | LSDU_Size = 46 |
| 64 | 4 | FCS | |

**Figure 10: Supervision frame format (untagged)**

| Offset | Length | 0 7 | 8 15 |
|---|---|---|---|
| 0 | 6 | PRP_DestinationAddress (multicast 01-15-4E-00-01-XX) | |
| 6 | 6 | PRP_SourceAddress (MAC address of the DAN) | |
| 12 | 2 | Type (0x8100 for VLAN) | |
| 14 | 2 | prio / cti | VLAN Identifier |
| 16 | 2 | Type (0x88FB for PRP) | |
| 18 | 2 | PRP_Ver | |
| 20 | 2 | PRP_TLV.Type = 20 or 21 | PRP_TLV.Length = 12 |
| 22 | 6 | PRP_SourceMacAddressA (MAC address A of the DAN) | |
| 28 | 6 | PRP_SourceMacAddressB (MAC address B of the DAN) | |
| 34 | 2 | PRP_TLV.Type = 30 or 31 | PRP_TLV.Length = 6 |
| 36 | 6 | PRP_SourceMacAddressA (MAC address A of the RedBox or VDAN) | |
| 42 | 18 | padding to 68 octets | |
| 60 | 2 | SequenceNr | |
| 62 | 2 | LAN = A or B | LSDU_Size = 46 |
| 64 | 4 | FCS | |

**Figure 11: Supervision frame format (tagged)**

| Parameter | | Description |
|---|---|---|
| PRP_DestinationAddress | | reserved multicast address 01-15-4E-00-01-XX (XX is “00” by default, but if conflicts arise, XX can be configured to take any value between 0x00 and 0xFF) |
| PRP_SourceAddress | | MAC address of the sending node |
| PRP_Ver | | protocol version, set to “0” (zero) for this version of PRP |
| first PRP_TLV entry (Type, Length, Value) | | |
| | PRP_TLV.Type | indicates the operation mode: Duplicate Discard (value 20) or Duplicate Accept (value 21) |
| | PRP_SourceMacAddressA<br>PRP_SourceMacAddressB | MAC address used by each port (these addresses are identical, except if address substitution is used) |
| second PRP_TLV entry (Type, Length, Value), used by RedBoxes only | | |
| | PRP_TLV.Type | indicates whether the supervision frame belongs to a RedBox (value 30) or a VDAN (value 31)<br>Remark: The DAN itself does not send supervision frames, but the corresponding RedBox does it as a proxy on behalf of all VDANs connected to it. |
| | PRP_SourceMacAddressA | MAC address A used by the respective RedBox or a VDAN |
| SequenceNr | | sequence number used for network supervision frames |
| LAN | | LAN over which this supervision frame is sent |
| LSDU_Size | | size of the LSDU (always 46, independent if tagging is used or not) |

**Table 2: Supervision frame fields**

## 5 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 state 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.

PRP defines an SNMP MIB for this purpose.

A network management tool is preferably a DAN and can access nodes and switches as if they all would belong to the same network. Especially, network management implemented in a DAN is able to see SANs connected to either LAN.

## 6 Configuration rules and parameters

Within a PRP network, the following rules should be followed:

- The network shall consist of two fail-independent LANs that have similar properties, i.e. each LAN is able to carry the traffic that would exist in the absence of redundancy.
- Both adapters A and B of a DAN shall be configured with the same MAC address.
- LAN_A plus LAN_B constitute an IP subnet. The IP addresses of all nodes and switches within this subnet shall be unique and share the same prefix. The redundancy scheme shall be transparent to the IP layer, i.e. a DAN shall have the same IP address when seen from either LAN_A or LAN_B.
- All nodes in the network shall be configured to operate with the same multicast address for the purpose of network supervision.
- All DANs shall be configured with the same LifeCheckInterval.

Table 3 describes the PRP parameters and their default value.

| Parameter | Description | Default value |
|---|---|---|
| LifeCheckInterval | how often the presence of a node is checked | 2 000 ms |
| NodeForgetTime | time after which node entry is cleared | 60 000 ms |
| LinkTimeOut | time after which a link is considered lost | 6 000 ms |
| DropWindowMax | max size of drop window | 32 768 |

**Table 3: PRP Parameters**

# 7 Implementing PRP

PRP can be implemented in different ways, depending on where duplicates are handled:

- On the driver level: This variant provides efficient handling transparent to the application.
- On the application layer: This variant is less efficient but easier to observe and debug. It is available as open source software under BSD-like license.
- In hardware: Such a solution does not consume computing resources of the host processor to run the discard algorithm.

The Institute of Embedded Systems (InES) of the Zurich University of Applied Sciences plays an active role in the area of real-time Ethernet since years. Its R&D activities result in a series of portable protocol implementations for real-time, synchronization, and redundancy functions.

With respect to PRP, the following basis is offered [6]:

## 7.1 Kernel mode variant of PRP software stack

A software PRP implementation may be designed as an intermediate layer that communicates with the upper layers by a virtual network device (PRP Driver). This layer presents towards its upper layers the same interface as a non-redundant network device (see figure 12).

By communicating over the PRP interface, the existence of the two real interfaces is hidden from the operation system and the upper layers. It intercepts the normal frame flow. It grabs all frames coming from the two interfaces and checks whether to forward the frames to the upper layers or not (discard algorithm). For the operation system, it seems like the frames are coming from the virtual PRP interface. In the sending path, the operation system will send the frames to the virtual PRP interface, which will duplicate the frames and send them to respective interfaces.

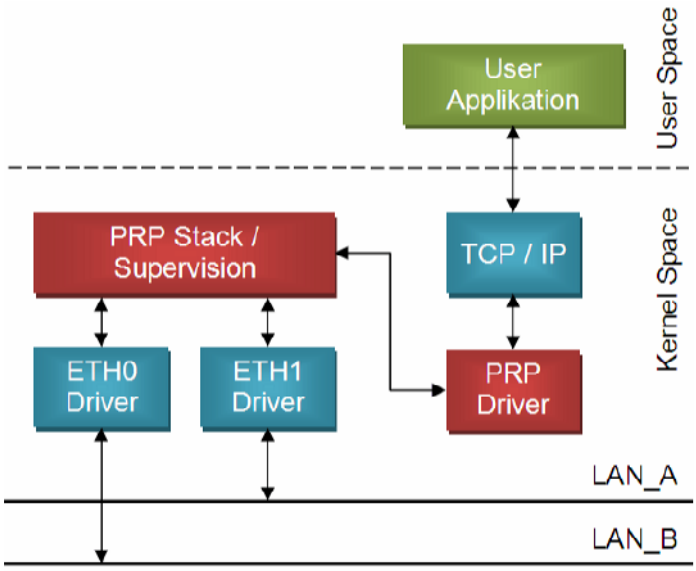



**Figure 12: Kernel mode variant of PRP software stack**

## 7.2 User mode variant of PRP software stack

For the evaluation of the PRP stack, an open source software is provided for Linux (see figure 13). The code covers the handling of the duplicates as well as the supervision functions. Thanks to the user mode execution the algorithms can easily be traced and tested.

Note: For operational use the kernel variant of the PRP stack is recommended for performance reasons.

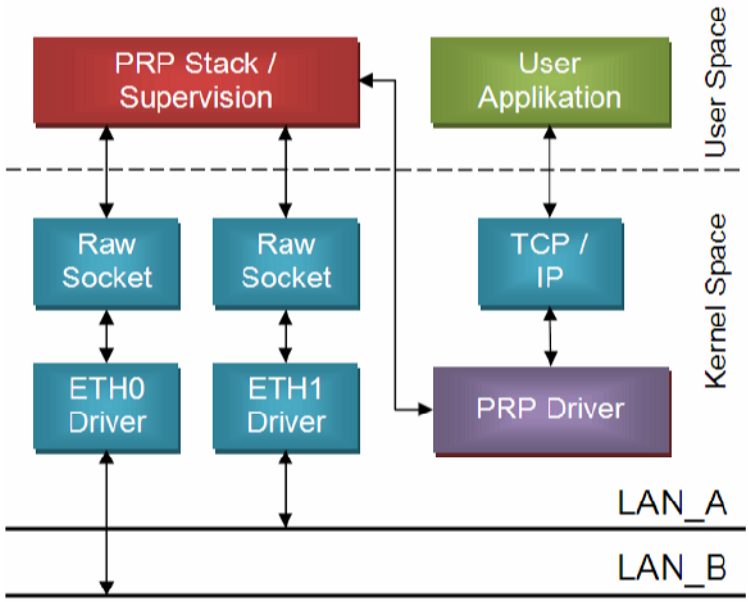



**Figure 13: User mode variant of PRP software stack**

## 7.3 FPGA-based implementation of a RedBox

The Redundancy Box (RedBox) is a three port switching device to connect conventional nodes (with only one network interface) to both LANs of a PRP network. It has one port to each LAN and one to the VDAN network

Because VDANs appear to the other nodes of the network like dual attached nodes (DAN) they are called virtual dual attached nodes (VDAN). The mechanism of duplicate generation und duplicate rejection is completely transparent to the VDAN. The RedBox completely mimics the nodes connected behind it and multicasts supervision frames on their behalf.

For management purposes the redundancy box itself is a DAN with an IP address of its own.

The implementation is an FPGA-based system on chip, containing three 100 Mbps duplex MACs, frame buffering and switching, and duplicate discard algorithm in hardware. Performance is at wire speed. For supervision and management functions, a processor core is included.

An alternative set-up is to use the RedBox code for a single VDAN implemented on the same device. Application code can be executed on the integrated processor core.

## 7.4 PRP redundancy manager

Spending redundant infrastructure to a network can significantly improve availability if failures are detected and repaired quickly. This requires an adequate redundancy monitoring tool. InES has developed such an SNMP management application with the following properties:

- Manager is connected to both LANs (i.e. is itself a DAN)
- Automatic discovery of network configuration (by observing supervision frames of DANs and VDANs and by scanning an IP address range to find SDANs)
- Support of all PRP defined node types and managed objects
- Graphical user interface provides overview of all nodes and their status in different views

**a) Views**

The general **overview** shows all detected and/or manually configured nodes as depicted in figure 14. The upper region shows SANs connected to LAN_A (with an inactive node “SAN 2”). The bottom region shows SANs connected to LAN_B. In between located are DANs, VDANs and their associated RedBoxes. “PRP Node 1” shows a failure in its interface to LAN_A and the LAN_A interface of the RedBox is in maintenance state.

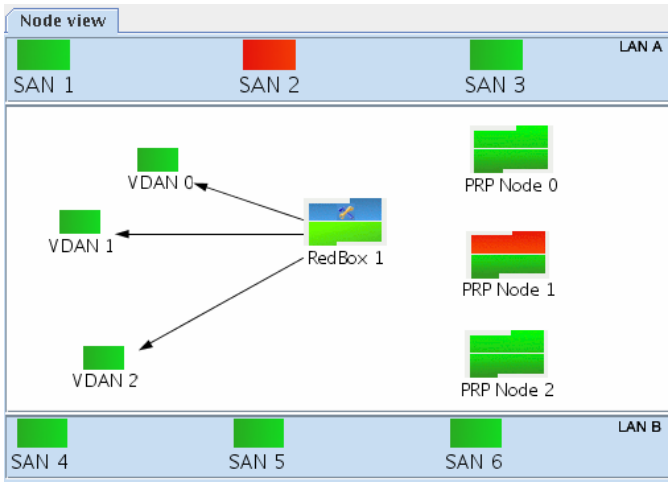



**Figure 14: General network overview**

A specific node can be selected to display this node’s view of the network. Figure 15 describes how “PRP Node 1” sees the network. Since it has failure in the interface to LAN_A it can only access DANs/VDANs with an operational interface to LAN_B plus SANs connected to LAN_B.

Zooming into a specific node allows to display all its values and properties (see figure 16).

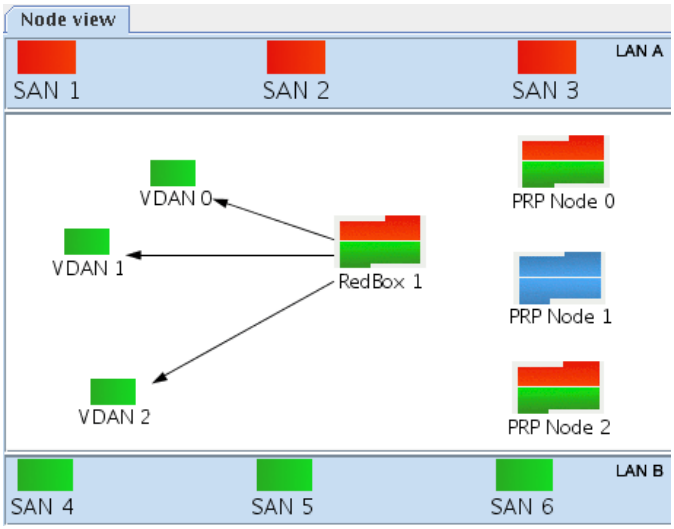



**Figure 15: Network view as seen by a selected node (the blue one is selected)**

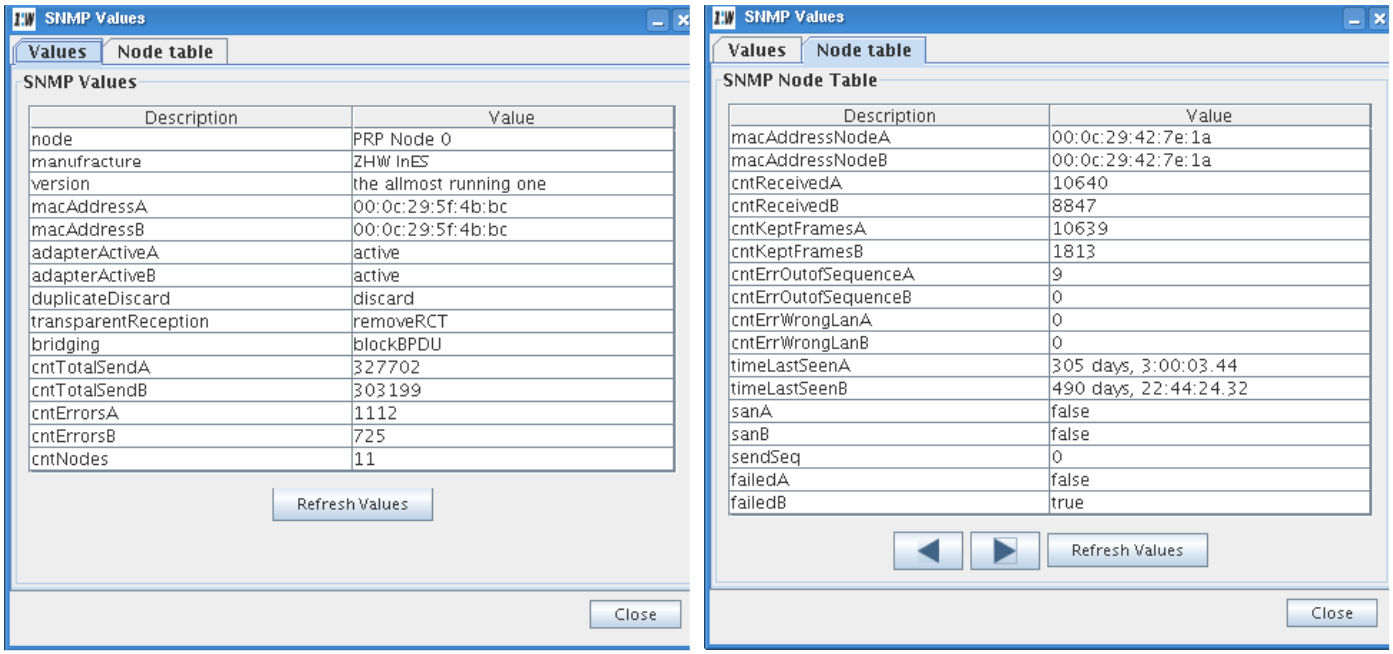



**Figure 16: GUI dialogues; DAN values (left) and nodes table (right)**

## b) Configuring manager and nodes

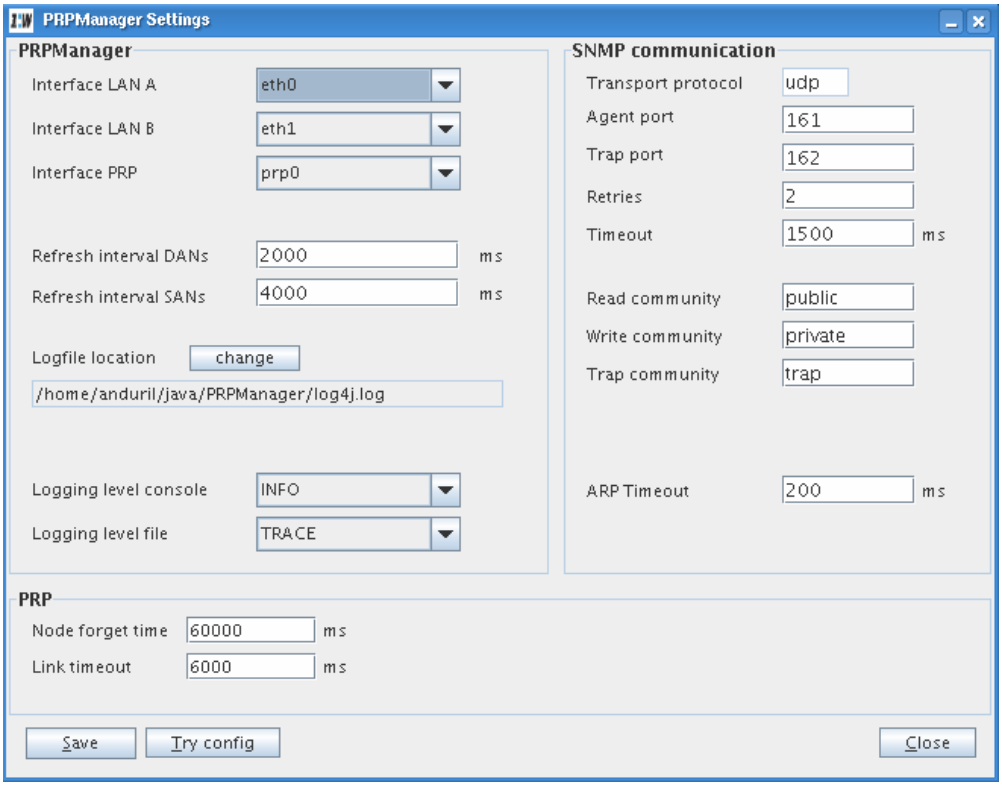



**Figure 17: PRP Manager settings Window**

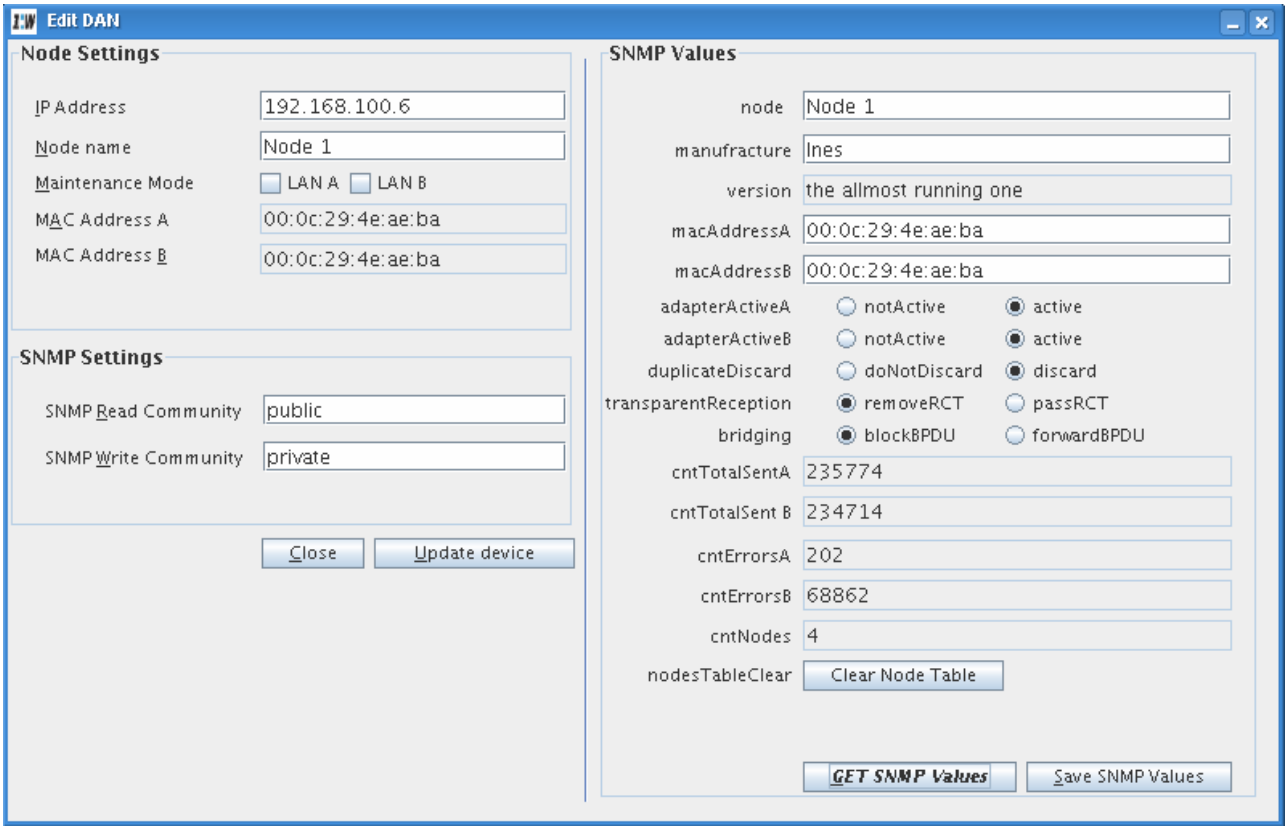



**Figure 18: DAN settings Window**

### c) Management Information Base (MIB) OID tree

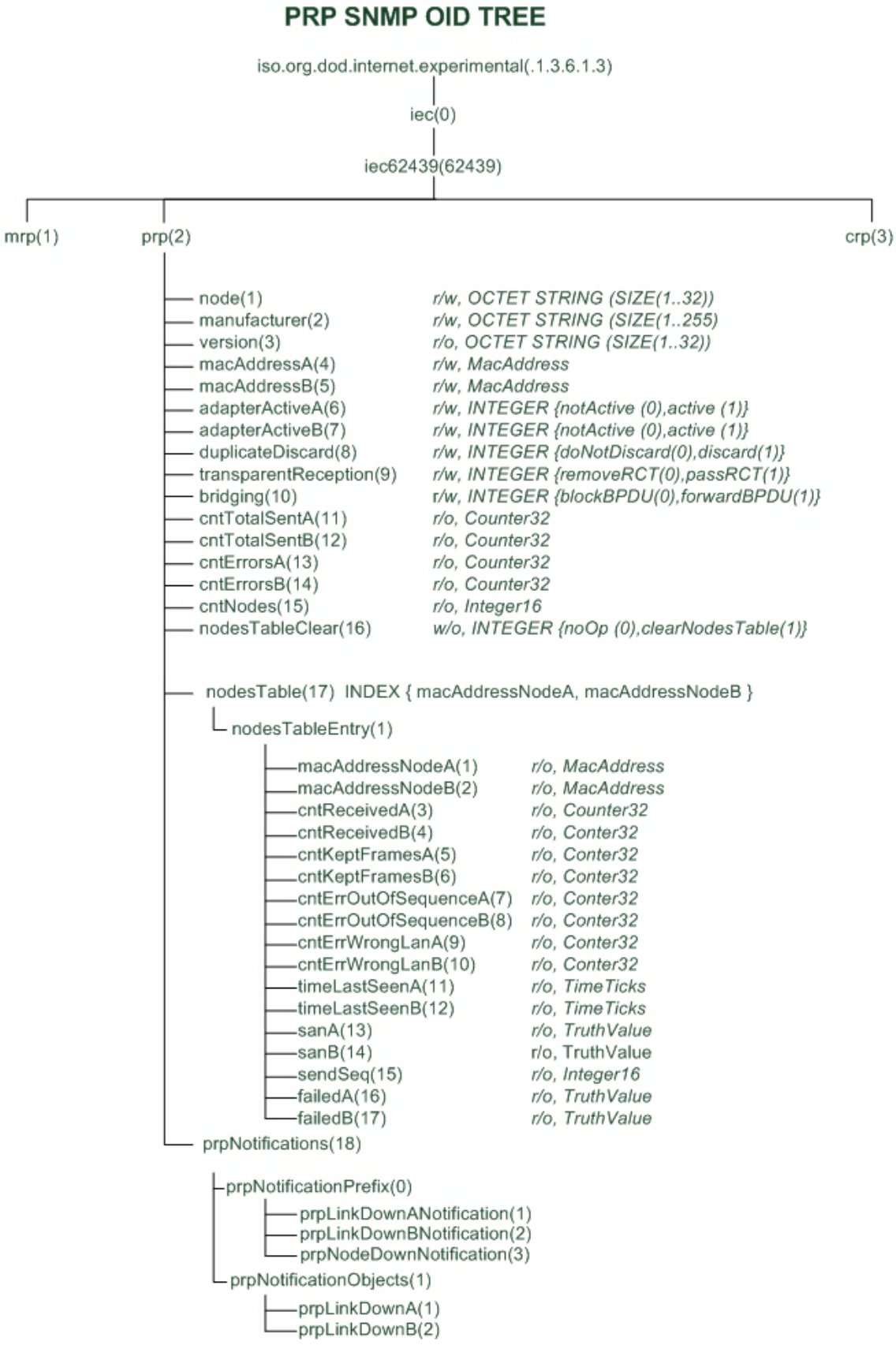



**Figure 19: PRP managed objects and their identifiers**

## 7.5 PRP combined with IEEE 1588 synchronization

High availability requirements often go together with real-time and synchronization requirements. The operational principle of the commonly used Precise Time Protocol (PTP) [4] is to measure the path delay of synchronization messages. Combining PTP with PRP leads to the problem that PRP hides the use of two networks to the upper layers. For PTP, this attempt is not permissible, because it matters on which network a path delay measurement takes place.

InES has proposed a solution to combine both PRP and PTP and has demonstrated a prototype on the occasion of the ISPCS 2007 in Vienna [5].

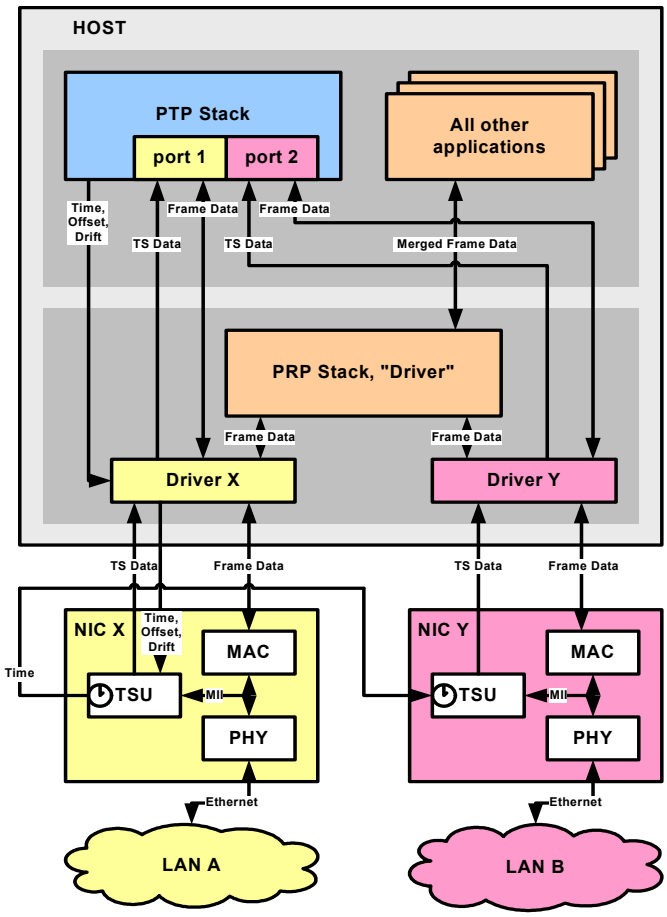



**Figure 16: Implementation of the multiport-clock model combines PRP with PTP**

The multi-port model was implemented on a Linux PC equipped with two IEEE 1588 enabled network interface cards (see figure 16). There is only one PTP instance running, but with two ports. In general, it behaves like a boundary clock, but with the specialty that it will never have one port in the slave and the other one in the master state. This means that the clock is actually running in slave only or master only mode. The detection of a failure is crucial but the IEEE 1588 defined Best Master Clock algorithm can be used for this purpose.

## Abbreviated terms and acronyms

| | |
|---|---|
| ARP | Address Resolution Protocol |
| DAN | Double attached node |
| FCS | Frame Check Sequence, IEEE 802.3 cyclic redundancy check |
| IP | Internet Protocol |
| LAN | Local Area Network |
| LSDU | Link Service Data Unit |
| MAC | Medium Access Control |
| MIB | Management Information Base |
| MRP | Media Redundancy Protocol |
| OID | Object Identifier |
| OUI | Organizational Unique Identifier, allocated by IEEE |
| PDU | Protocol Data Unit |
| PRP | Parallel Redundancy Protocol |
| PTP | Precision Time Protocol, according to IEEE 1588 [4] |
| RCT | Redundancy Control Trailer |
| RSTP | Rapid Spanning Tree Protocol, according to IEEE 802.1D [3] |
| SAN | Single Attached Node |
| SNMP | Simple Network Management Protocol |
| VDAN | Virtual Doubly Attached Node |
| VLAN | Virtual LAN, according to IEEE 802.1Q |

## References


[1] IEC 62439, Committee Draft for Vote (CDV): “Industrial communication networks: high availability automation networks”, chapter 6, entitled Parallel Redundancy Protocol, April 2007.

[2] IEC 62439, Committee Draft for Vote (CDV): “Industrial communication networks: high availability automation networks”, chapter 5, entitled Media Redundancy Protocol based on a ring topology, April 2007.

[3] IEEE 802.1D-2004, “Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges”, chapter 17, June 2004.

[4] IEEE P1588/ D1, “Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems”, June 2007.

[5] Hans Weibel, Sven Meier “IEEE 1588 applied in the environment of high availability LANs”, International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, October 2007, Vienna

[6] http://ines.zhaw.ch/prp