2012 IEEE International Symposium on Industrial Electronics, Hangzhou, China, 2012, pp. 1197-1202, doi: 10.1109/ISIE.2012.6237259

High Availability Automation Networks: PRP and HSR Ring Implementations

J.A. Araujo, J. Lázaro, A. Astarloa, A. Zuloaga
Electronics and Telecommunications department
University of the Basque Country (UPV/EHU)
Bilbao, Spain

A. García
SoC-e (System-on-Chip engineering, S.L.)
Zitek Bilbao (ETSI)
Bilbao, Spain

***Abstract*— The work developed has as basis the networks/protocols described in the standard IEC 62439-3 Industrial automation networks - High availability automation networks: PRP and HSR. The similarities of both networks and a software implementation over Linux of PRP protocol have been the starting points taken for this work. A prototype of a HSR node has been developed; this prototype was proved over some virtual machines connected in a ring network as the standard states and achieved the target of seamless availability. Then it has been ported over an FPGA to validate the system in a FPGA/PC heterogeneous network.**

I. Introduction

In many industrial applications it is increasingly necessary to have networks with zero recovery time when an error occurs in the network. There are different methods in which, with some redundancy, the communication is recovered from a fault in the network; this recovery takes some time, and even if it is short, it could be unacceptable for some applications, for example in electric substations. To find out a solution, IEC 62439 series studies and proposes several alternatives of redundancy but only two of them provide zero time recovery from a fault in the network. More concretely the IEC 62439-3 [1] presents these two solutions: Parallel Redundancy Protocol (PRP) and High Availability Seamless Redundancy (HSR).

These two proposals are compatible with the Ethernet standard IEEE802.3 [2] and they are based on the introduction of a Link Redundancy Entity (LRE) which manages the redundancy included by the methods, described in the following sections, in a transparent way to all upper OSI layers.

This article firstly analyzes the state of the art, explains the redundancy solutions proposed in IEC 62439-3 for high availability seamless redundancy and actual implementations. Then comparisons, measurements and implementation work are presented and, at the end, conclusions and future work can be found.

II. State Of The Art

The standard IEC62439 series considers two different classes of network redundancy [3], [4]:

- Redundancy managed within the network
- Redundancy managed in the end nodes

Redundancy managed within the network offers redundant links and switches, but nodes are singly attached to the switches through no redundant links. Only part of the network is redundant and is normally inactive, so that requests some insertion delay.

Redundancy Managed within the end nodes adds several communication links to the end nodes. Parallel networks or paths provide high availability seamless recovery for those applications that need it and, in this case, the unique non-redundant elements are the nodes themselves.

That kind of methods for industrial automation networks have been studied for last years [3], [5], [6]. In this work, two of the most promising parallel network approaches collected in IEC 62439-3 standard are covered, these are PRP and HSR [1].

A. Parallel Redundancy Protocol (PRP)

PRP obtains high availability introducing two networks which function in parallel. Figure 1 shows a basic PRP network topology. The main elements are the nodes linked to two networks: Doubly Attached Node with PRP (DANPs), which duplicate information and send it through the two different links. If any fault happens in one of the networks the communication continues over the other one without loss of information. Exchange for this availability, two networks must be maintained and managed; besides, duplicated information must be handled and discarded. For discard duplicates the standard proposes an algorithm, but doesn't discard other possibilities.

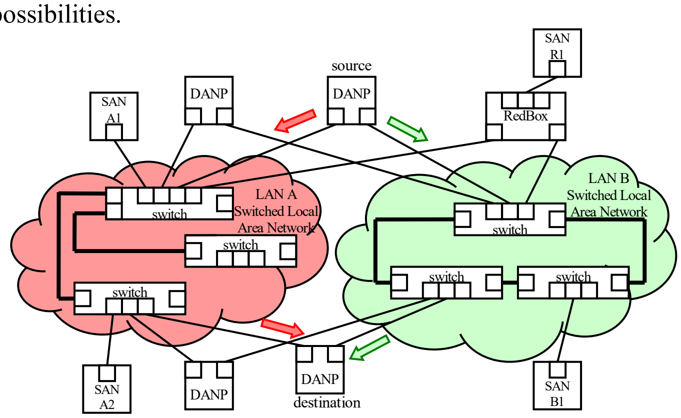


Figure 1. PRP network example.

In a PRP topology, nodes connected only to one of the networks are accepted. These are named SAN (Single Attached Nodes), but these nodes can only interchange information with nodes connected to the same network. Other possibility is to connect this kind of node to the two networks, using a RedBox (Redundancy Box) which manages frames used in the PRP network.

PRP is based on redundancy at OSI level 2 (Link). A new layer is introduced in the communication stack, LRE. This entity manages all functions, duplicates and discard algorithms, so that for upper layers PRP works as a simple Ethernet interface.

In order to achieve its functions PRP adds some octets to the link frame. This new field is called RCT (Redundancy Control Trailer).

Figure 2 represents the PRP frame with the RCT appendix attached. It is compound by Sequence Number, LAN Identifier and LSDU Size [1]. For SANs, this RCT is simple padding.

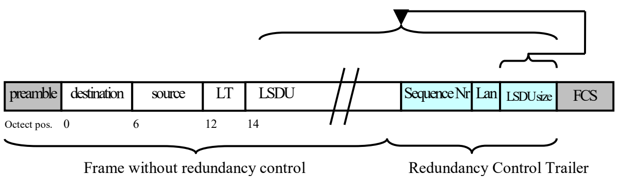


Figure 2. PRP frame with PRP trailer.

B. *High Availability Seamless Redundancy (HSR)*

HSR overcomes one of the main disadvantages that PRP suffers. The idea is achieving the same redundancy result but using a unique network. In this case, there is one network but two paths from source to destination.

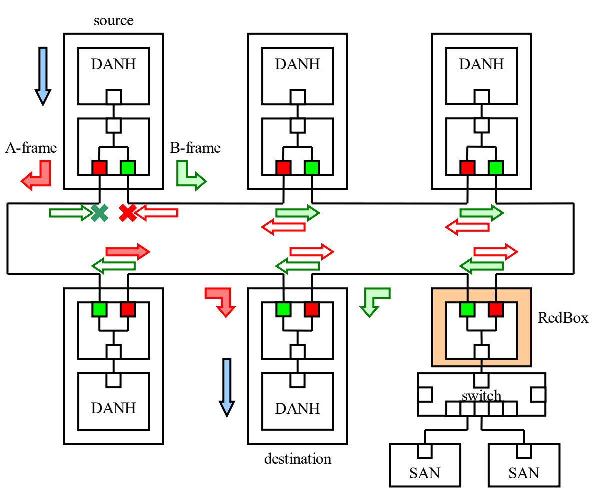


Figure 3. Unicast envy in a HSR network.

The basic network topology of HSR is a ring, Figure 3 presents this topology. The network nodes are identified as Doubly Attached Switching Nodes with HSR (DANHs). Each one is connected to the two nodes next to it so that in the same way as in PRP, frames are duplicated and sent over different paths.

HSR does not admit SANs because those can not resend frames and can not be connected in both directions. If a SAN need to be connected to the network HSR RedBox, which implements the functions needed, shall be introduced.

HSR also applies redundancy at layer 2. A specific LRE is defined for HSR and implements HSR functions of managing frames, forwards, duplications, discards, etc. thus HSR is transparent for upper layers [5] and is compatible with the Ethernet standard.

Now instead of a trailer (RCT), LRE adds a HSR tag composed of 6 octets, this tag is inserted just after addresses in the frame allowing the localization of a HSR frame by finding the correct EtherType. Figure 4 represents HSR frame with HSR tag.

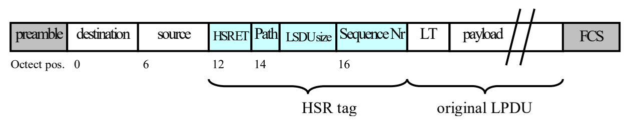


Figure 4. HSR frame with HSR tag.

The standard contemplates not only the use in rings but also in other network structures, as for example with parallel networks, the interconnection between PRP and HSR networks, interconnection between more rings and rings of rings using QuadBoxes [1], [6]. A QuadBox has 4 ports and makes it possible to interconnect two HSR Rings using 2 ports for each ring. It manages frames and duplicates of each ring and resends them to the other one. In this configuration the QuadBox conducts the traffic, but to avoid single point failure vulnerability rings should be connected in two different points using two of them.

The discard algorithm must take into account circulating frames [1]. Those frames could appear when a multicast frame lost its source, or when an unicast frame lost source and destination. In these cases frames would continue to circulate around the ring. This situation must be avoided, so that more algorithms are being studied [7].

In applications which time delays are critical, HSR should work in cut-through mode to reduce delay. In this situation, after destination field, source address and sequence number are received and the frame is confirmed to be forwarded the node begins forwarding the frame over the other line [1], [7].

Another remarkable feature of HSR and PRP is that multicast supervision frames are send by every nodes so that all the nodes have an updated table of the network [1].

C. *Current Implementations*

There are implementations and groups studying and developing different parts of the standard.

The InES, Institute of Embedded Systems [8] of the Zurich University of Applied Sciences (ZHAW) offers some solutions for PRP. PRP Software stack which communicates upper layers with two physical links by a virtual network device. PRP manages

Ethernet frames generating and discarding duplicates, adding and removing RCT trailer and, on the other hand it supervises networks and physical redundancy. InES offers this PRP implementation for Linux in User Space and in Kernel Space. This group has developed an IP core (Intellectual Property core) with RedBox functionality. This IP core is designed for Altera FPGA and uses Linux over a soft-processor. For evaluation purposes a board suitable for PRP RedBox is offered too [9].

In the field of HSR InES is working in a design of an HSR RedBox to be offered as an IP core like with PRP. As commented before new algorithms have been studied in [7] to manage duplicated and circular frames obtaining some simulations and successful results in tests made.

Ruggedcom [10] offers the development of PRP and HSR to clients in its web.

Lattice Semiconductor Corporation [12] and Flexibilis Oy [13] offers an FPGA Ethernet switch IP cores with HSR support. Those IP cores are for Lattice FPGAs (LatticeEPC3). They announced the immediate availability of the Flexibilis Ethernet Switch (FES) IP cores.

Hirschmann [15] (a Belden [16] brand) offers RSP switches optimized for the type of data communication used in the electricity supply industry, usable too in many other areas. Those switches support PRP and HSR.

III. Reliable Ethernet Implementations on Virtual Machines

The implementations of HSR and PRP networks have been done using virtual machines. Those virtual machines have been implemented over VirtualBox 4.0.4 and use GNU-Linux (Ubuntu 10.04 and 10.10) operating system. Wireshark 1.2.7 has been used to analyze frames interchanged between the machines and to see duplications, forwards, HSR tags, RCT trailers, supervision frames, etc.

A. HSR Implementation

In order to build a suitable test framework a HSR network is implemented in some virtual machines connected in a ring. Every machine has two Ethernet ports connected to the previous one and the other to the next one in the ring; the last one is connected to the first one, thus closing the ring.

The basic setup is depicted in the Figure 5. The ring is composed of three machines:

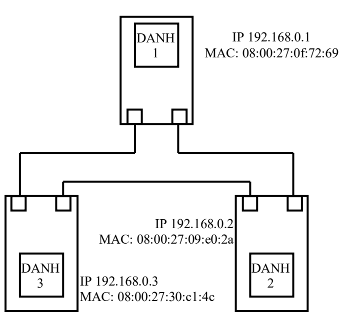


Figure 5. HSR ring implemented using virtual machines.

The implemented network has the basic functionalities of an HSR network and achieves the high availability seamless communication between the machines. In order to do that it uses a software solution which manages frames in the link layer:

- Frames are properly sent and received.
- Duplicated frames are properly removed in the destination.
- Multicast frames are resent by every nodes unless the source one.
- Communication is not interrupted if one of the nodes between source and destination falls down, achieving the issue of high availability of this kind of network.

Figure 7 shows the frame flow interchange by one of the machines, DANH1, with the other two machines next to it through its two links when machine 1 makes a ping to 3 and the ring functions properly. In “b” and “c” the same frame is selected, it is a ping request and has the HSR Tag with the same sequence number, in “a” the same frame delivered by upper layer without HSR tag is selected. Just below this frame in three cases is the reply, which is received through two links but only one of them, of the two ports, is delivered to upper layer after removing HSR tag, which appears in every frames of “b” and “c”.

On the other hand, in two links, “c” and “b”, some packets appear which are not passed to upper layers. Those multicast packets (destination 01:15:4e:00:01:00 as standard states) are supervision frames, in each link appears two per node in the network, one on each direction.

The key feature of HSR is network redundancy. In the following experiment, the link between nodes 1 and 3 fails as sketched in Figure 6. :

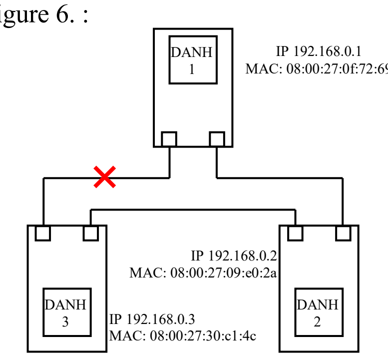
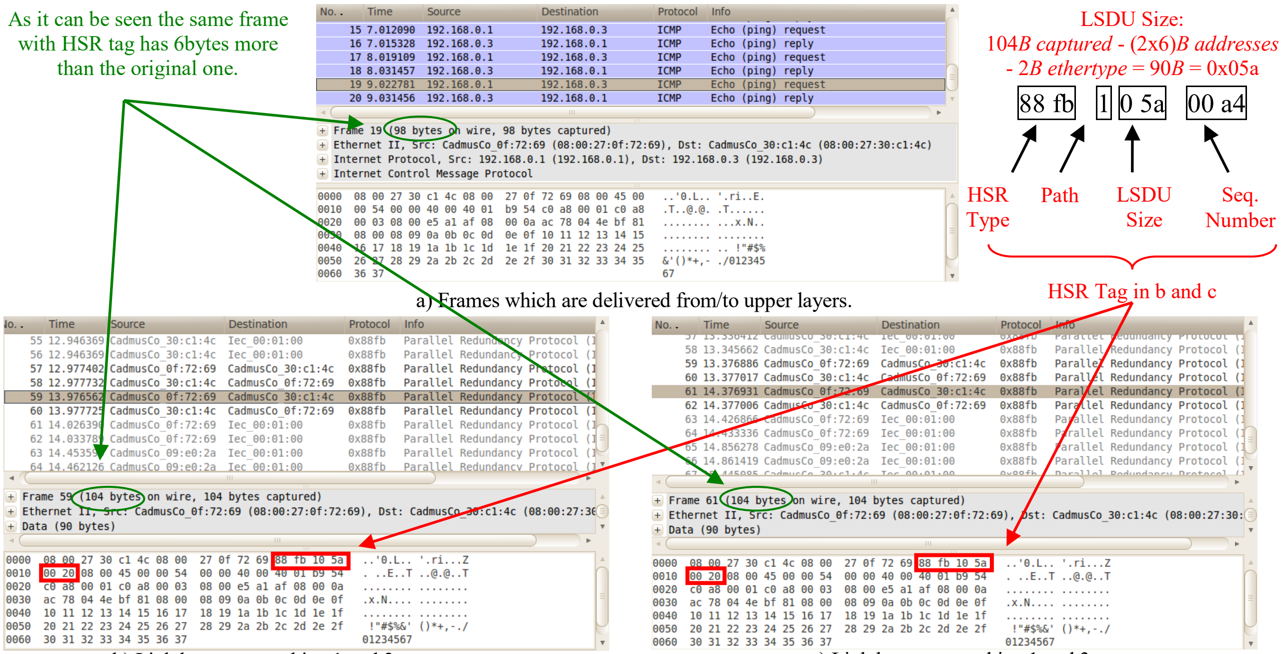


Figure 6. HSR ring implemented using virtual machines. Link between 1st and 3rd machine is opened.

The results of this experiment are summarized in Figure 8. The same ping request packet has been made. The reply to the ping appears only in one of the links, the one connected to the second machine (DANH2) “a”, through which the communication flows. The fault in the ring is transparent for upper layers. In two cases the ping request/reply is made perfectly and upper layers do not know what happen in lower layers, the behavior for upper layers is the same in both cases (Figure 7a) with and without fault.

As it can be seen, in Figure 8, only one supervision frame per node appears in each link, it is because in each direction only flows one frame and doesn’t continue because of the broken link.

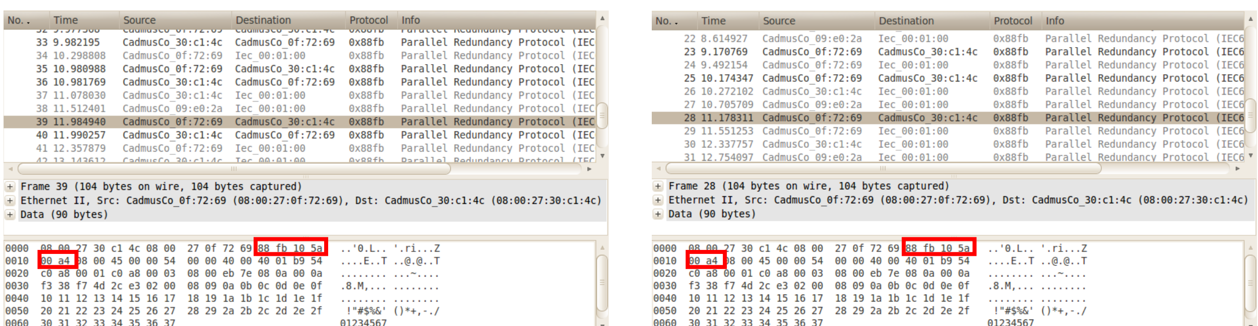


a) Frames which are delivered from/to upper layers.

b) Link between machine 1 and 2.

c) Link between machine 1 and 3.

Figure 7. Frames interchanged by DANH1 through two links in HSR ring.



a) Link between machine 1 and 2. Requests and replies can be observed.

b) Link between machine 1 and 3 broken. There is no reply.

Figure 8. Frames interchanged by DANH1 through two links in HSR ring when one link in the ring is broken (between 1 and 3).

B. *PRP Implementation*

An equivalent set up has been made with a PRP network. In this case, three virtual machines were been connected to two parallel networks. The first machine DANP1 made a ping to DANP3 that replies through 2 LANs. This setup is presented in Figure 9.

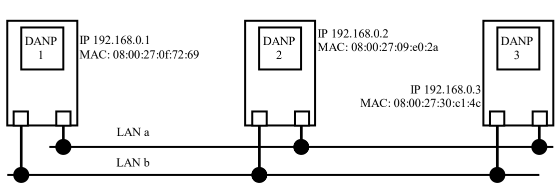


Figure 9. PRP network implemented using 3 virtual machines.

In Figure 11, ping requests and replies can be observed in each link but only one is delivered from/to upper layers.

After this ping operation, the link of the DANP3 to the LAN b is broken, Figure 10. In upper layers there is no variation (Figure 11a) but if the link of machine 1 to LAN b is analyze it can be seen that there are only requests but not replies, as it can be notice in Figure 11.

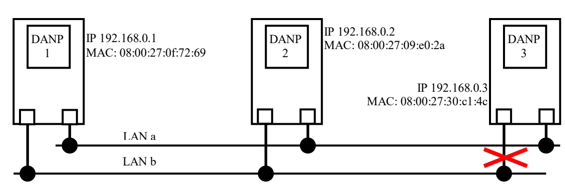
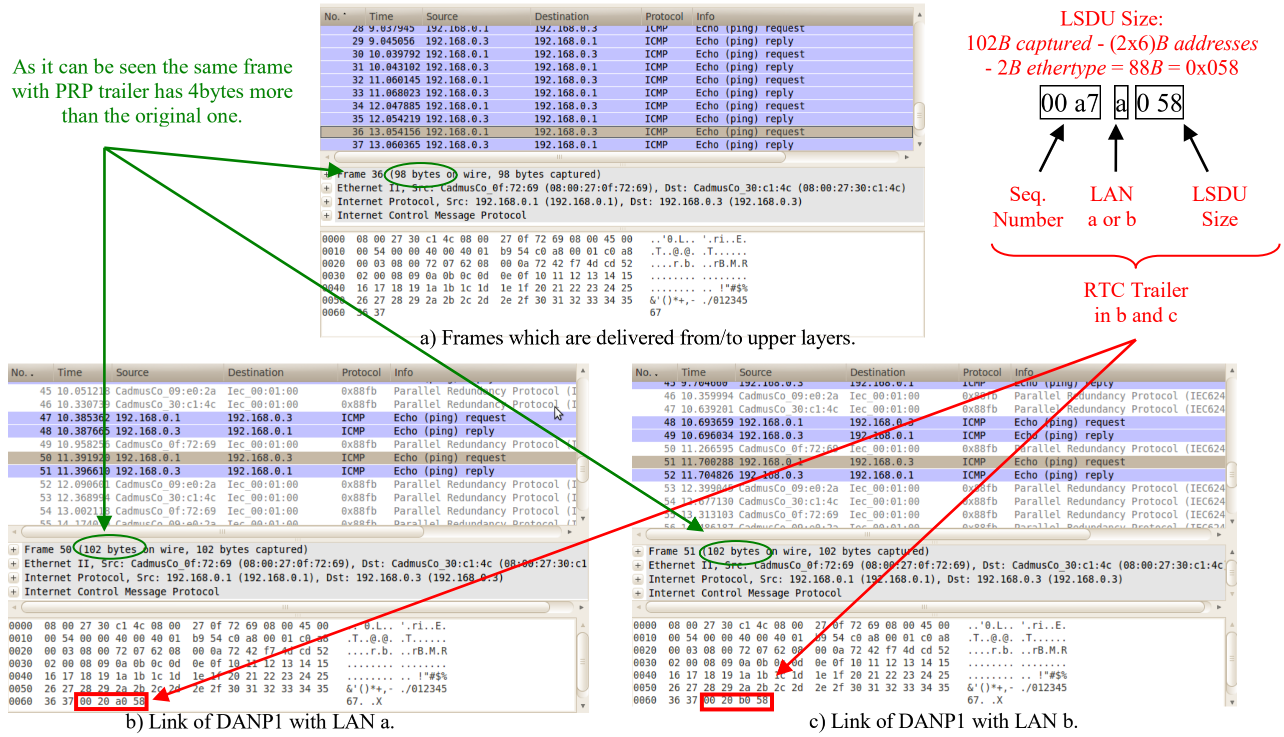


Figure 10. PRP network implemented using 3 virtual machines. Link between DANP3 and LAN b is broken.

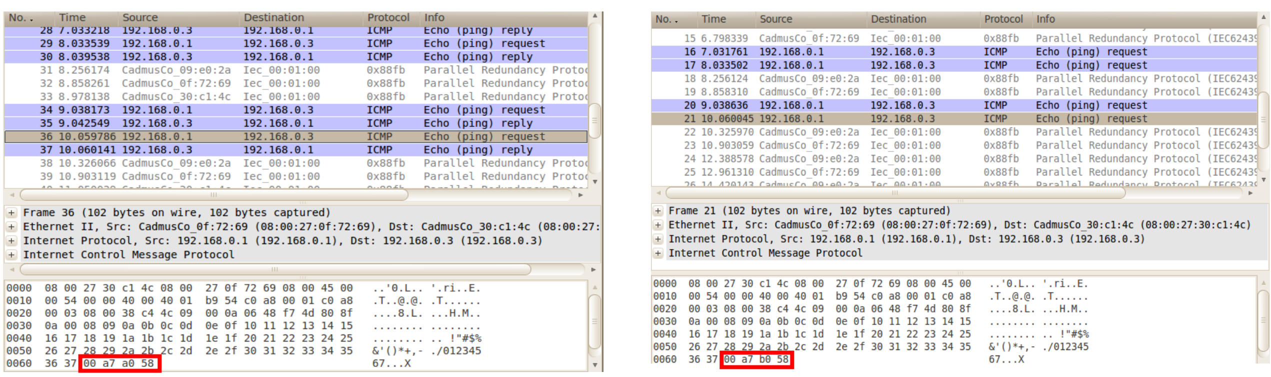


a) Frames which are delivered from/to upper layers.

b) Link of DANP1 with LAN a.

c) Link of DANP1 with LAN b.

Figure 11. Frames interchanged by DANP1 through two links in PRP network.



a) Link of DANP1 with LAN a. There are requests and replies.

b) Link of DANP1 with LAN b. Link of DANP3 with LAN b broken. There is no reply.

Figure 12. Frames interchanged by DANP1 through two links in PRP network when the link of DANP3 with LAN b is broken.

C. Comparisons

In both cases, PRP and HSR over virtual machines, communication is maintained in a transparent way for upper layers even if a failure in one link happens, as it was expected. So that, both software implementations provide high availability seamless communication against a single failure in the network.

IV. Reliable Ethernet Implementation Of HSR On Xilinx FPGA

Once the set up with virtual machines running in PC machines has shown the expected behavior, HSR software has been compiled for Petalinux 1.4 [11] to be run in a System-on-Programmable-Chip (SoPC) with a soft-processor MicroBlaze. This SoPC is implemented in a Spartan Xilinx FPGA, specifically XC6SLX45T. The evaluation board used for the evaluation is the SP605. In order to give two new Ethernet ports needed a module with two Ethernet ports has been added (ISM Networking FMC Module).

PetaLinux is a System Development Kit (SDK) specifically targeting FPGA-based System-on-Programmable-Chip designs for Embedded Linux on Xilinx FPGAs. Containing tools to build, develop, test and deploy Embedded Linux on FPGA projects of programmable logic and Embedded Linux.

The design developed with Xilinx SDK 12 software for the FPGA includes the following elements:

- Microblaze: A Xilinx 32-bit RISC Harvard architecture soft processor core with a rich instruction set optimized for embedded applications. [14].
- Digital Clock Manager (DCM): The Clock Generator provides clocks according to system wide clock requirements.
- MicroBlaze Debug Module (MDM): Enables JTAG-based debugging of one or more MicroBlaze processors.
- 3 Ether Lite MAC: One to communicate the FPGA with a PC to load the program to microblaze, and another two to implement HSR communication.
- RS232 UART: To provide RS232 communication.

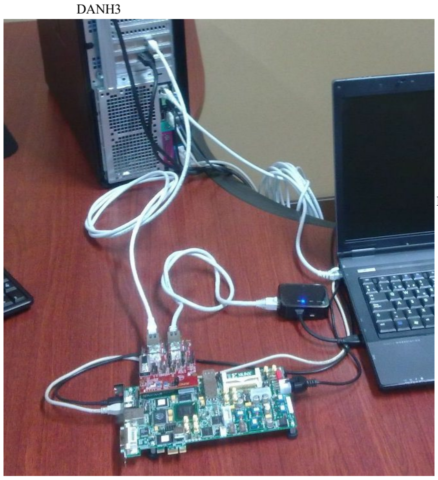
The entire system takes up the resources summarized in Table I.

TABLE I. HSR BASIC IMPLEMENTATION ON A SPARTAN-6 FPGA (XC6SLX45T)

Resource type	**Resource utilization**
Slice LUTs	6729 (24%)
Slice Registers	6177 (11%)
Occupied Slices	2841 (41%)
16B BlockRAM	17 (14%)

This FPGA running has been probed with two PC running the same HSR software, looking results in one of the computers with Wireshark and having similar results as with virtual machines. The setup (Figure 13. is similar to the one seen with virtual machines, now DANH1 and DANH2 are 2 different PCs and the third one, DANH3, is an FPGA running the HSR software in an embedded system.

DANH3



DANH1

DANH2 (FPGA)

Figure 13. HSR ring implemented using 2 PCs and an FPGA running the HSR software.

When DANH1 made a ping to DANH2 alternatively the links DANH1-DANH2 and DANH1-DANH3 where disconnected and ping replies where maintained, that is to say, communication was maintained in a transparent way for upper layers in spite of broken links.

V. CONCLUSIONS AND FUTURE WORK

The comparison of PRP and HSR experiments over virtual machines shows how two solutions get the high availability of the communication with zero time recovery.

The step given in this work with the HSR implementation would be the basis for a complete implementation of the whole HSR standard and coming new version. The basic functions of the HSR have been developed, and tested. For future works there should be implemented and added more functions of the standard, as deleting circulating frames, to obtain a better and whole HSR software implementation. With this software version as basis Redboxes could be developed to function over an FPGA and easily connect a SAN to an HSR network.

A software solution in a PC or into an FPGA could be appropriate for those applications in which time is not critical, but availability is. For those applications where time must be taken in concern some tasks should be taken out from software to execute on hardware to obtain better rate results as the standard states with the cut-through mode.

ACKNOWLEDGMENT

This work has been carried out inside de Research and Education Unit UFI11/16 of the UPV/EHU and supported by the Department of Education, Universities and Research of the Basque Government within the fund for research groups of the Basque university system IT394-10 and the the research program SAIOTEK (project S-PE11UN059).

REFERENCES

[1] IEC 62439-3 (Ed.1.0): Industrial Communications Networks – High availability automation Networks – Part3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR), February 2010.

[2] IEEE Std 802.3 – 2005, Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, IEEE. (Ethernet). http://standards.ieee.org/about/get/802/802.3.html

[3] IEC 62439-1 (Ed.1.0): Industrial Communications Networks – High availability automation Networks – Part1: General concepsts and calculation methods, February 2010.

[4] Hubert Kirrmann, Mats Hansson, Peter Müri, "IEC 62439 PRP: Bumpless Recovery for Highly Available, Hard Real-Time Industrial Networks". ETFA Sept. 2007.

[5] Hubert Kirrmann, Oliver Kleineberg, Karl Weber and Hans Weibel, "HSR: Zero recovery time and low-cost redundancy for Industrial Ethernet (High availability Seamless Redundancy, IEC 2439-3)", ETFA Sept. 2009.

[6] David Gunzinger, Hans Dermot Doran "HSR Perfomance Evaluation: a pre-Specification Software Implementation", ETFA Sept. 2009.

[7] Xiaozhuo Jiang, "High Availability Seamless Ring Protocol Implementation in FPGA", Master Thesis, Swiss Federal Institute of Technology Zurich, 2009.

[8] InES (Institute of Embedded Systems): http://www.ines.zhaw.ch

[9] InES (Institute of Embedded Systems) solutions for PRP: http://www.ines.zhaw.ch/en/engineering/ines/high-availability/prp/solutions.html

[10] RudggedCom: http://www.ruggedcom.com/applications/hsr-prp/

[11] Petalinux: http://www.petalogix.com/products/petalinux/petalinux

[12] Lattice Semiconductor Corporation: www.latticesemi.com

[13] Flexibilis Oy: www.flexibilis.com

[14] Xilinx microblaze: http://www.xilinx.com/tools/microblaze.htm

[15] Hirschmann: www.hirschman.com

[16] Belden: www.belden.com