Duplicate and Circulating Frames Discard Methods for PRP and HSR (IEC62439-3)

J.A. Araujo, J. Lázaro, A. Astarloa, A. Zuloaga, N. Moreira
Department of Electronics Technology
University of the Basque Country UPV/EHU
Bilbao, Spain

Abstract— Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR), defined in the IEC62439-3 about high availability automation networks and proposed as reference network topologies for substation communication networks by IEC61850, assure the upkeep of the communication when an error in a network occurs. Those methods consist in sending information duplicated through different and independent paths; so that nodes must be capable to eliminate the duplicated information circulating in the network. This paper analyzes ways to do the discard of those frames, which is crucial for a good implementation of the standard.

Keywords— Ethernet; availability; redundancy; HSR; PRP; duplicate; circulating frame, IEC62439-3

I. Introduction

Ethernet is increasingly being used in different fields where a communication network is needed, but it supposes different requirements depending on the application. One of the possible requirements is the availability, and the way to increase it is adding redundancy.

The IEC62439 series presents different redundancy methods depending on the recovery time [1]. There are two protocols that offer zero time recovery which are collected in the IEC62439-3 Industrial communication networks – High availability automation networks – Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR) [2]. Those networks have been also considered as reference for Station and Process buses for substations as stated in the IEC61850-90-4: Communication Networks and Systems in Substations - Part 90-4: Network Engineering Guidelines [3].

This paper firstly introduces the basic concepts of PRP and HSR protocols needed, as operation, frames identification and time considerations. Then it is focused in the proposition, analysis and optimizations of different ways to eliminate duplicated and circulating frames, critical aspect to implement the standard.

II. State of Art

Both PRP and HSR provide bumpless redundancy, high availability, by introducing more than one independent path to send information duplicated from source to destination, so that in case of error no information is lost. In return duplicated frames that arrive to destination and circulating frames (those that circulate around the HSR network) must be eliminated.

PRP introduces redundancy using Doubly Attached Nodes (DAN) linked to two parallel Local Area Networks, LANs. Source duplicates frames and sends them through the two LANs; on the other side the destination node receives frames duplicated as sketched in Figure 1. The first frame that arrives is received and the one that arrives afterwards is discarded.

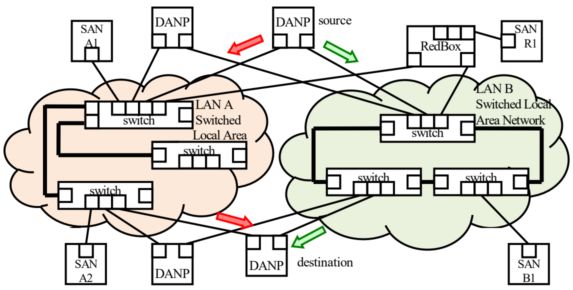


Figure 1. PRP network.

HSR nodes have two ports too, Doubly Attached Nodes with HSR. A simple HSR network consists in HSR nodes connected in a ring topology as depicted in Figure 2. The source sends information in the two directions and frames arrive to destination having traveled different paths. In HSR apart from duplicates, circulating frames could appear; those frames give a whole round to the network and they are received again in the same direction.

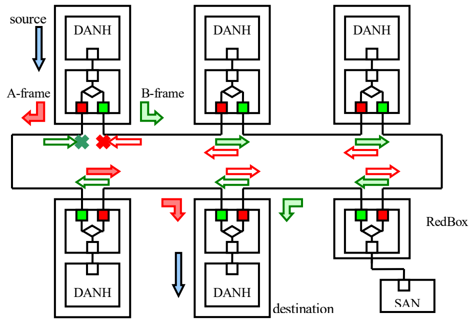


Figure 2. HSR Network.

The reception of duplicates would not be a problem because transport layer is able to manage them, but both PRP and HSR include redundancy at link level making redundancy transparent for upper layers and unloading processor from those tasks. A Link Redundancy Entity (LRE) is included and is the element which manages PRP/HSR functions [2].

PRP accepts Single Attached Nodes (SAN) directly connected to one of the LANs, but another way to connect a SAN is using a Redundancy Box (RedBox) which permits SANs to be connected to the two LANs (see Figure 1). Using RedBoxes is the mandatory way to connect SANs to an HSR network in which nodes not only receive frames but also must forward them (see Figure 2). RedBoxes are used to interconnect PRP and HSR networks too, as sketched in Figure 12.

III. Frames Identification

Both in PRP and HSR, the way to know if a frame has been seen before is remembering the frames that have arrived to nodes, and in order to do that frames must be identified. Those protocols add extra information to Ethernet frames, and using it and addresses of the frame, frames are identified.

PRP adds a trailer, Redundancy Control Trailer (RCT), at the end of the frame as depicted in Figure 3.

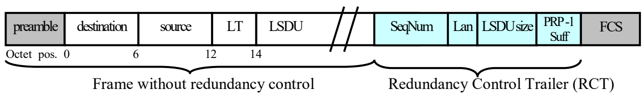


Figure 3. PRP frame.

And HSR adds a HSR Tag after addresses as Figure 4 shows.

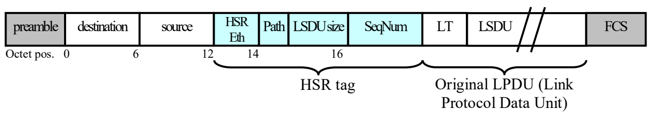


Figure 4. HSR frame.

Some fields are analogous and other equal [2]:

- Sequence Number (16bits): Every node has a sequence counter whose value is incremented every time a PRP/HSR frame is sent. Thus, the field Sequence Number is filled with the value of that counter each time a frame is duplicated and added trailer/tag to be sent. So that, the two copies sent have the same sequence number.
- LSDU size (12bits): Size (octets) of the Link Service Data Unit.
- LanId / PathID (4bits): In PRP, LanId indicates the network through which frame is sent. In HSR, PathId identifies the HSR regular frames direction.
- PRP-1 Suffix / HSR Ethertype (16bits): PRP-1 field (0x88FB) appears since second edition of the standard. HSR Ethertype (0x892F) is the own type for HSR protocol.

Then, the sequence number is common to both frames generated from each original frame delivered from upper layers to send through the two different paths by each node. Thus, those frames can be identified by using source sddress and sequence number, so that, nodes can store this information to discard duplicates and circulating frames.

Standard does not specify any discarding method. It only states that the chosen algorithm has to guarantee that a legitimate frame must never be rejected, while occasionally a duplicate can be accepted. It lets in the implementer's hands the use of node tables, hash tables, FIFOs or sequence number tracking. Below, various methods, using those mechanisms to discard frames, are going to be analyzed. Besides, the standard presents some time requirements for discarding frames which have to be taken into account.

IV. TIME CONSIDERATIONS

The IEC62439-3 [2] presents some time considerations in order to help managing duplicates for PRP. They can be adapted and applied to duplicates and circulating frames for HSR too.

Figure 5 shows how frames arriving through a port after having arrived through the other one are considered duplicates or not.

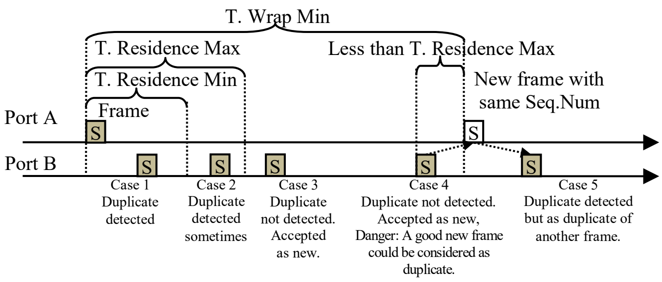


Figure 5. Time considerations to detect duplicates and circulating frames.

T. Residence is the time that information about a frame is remembered so as to check if new frames are duplicate/circulating frames or not. It could be fix or a margin (Max-Min) depending on the method used. It is smaller than T.Wrap Min, the minimum theoretical time in which a source can repeat a sequence number (400ms at 100Mbps).

In general, cases 4 and 5 (see Figure 5) would be avoided if the network is constructed with a minimum care of not having delays too long. For example, the time that it takes to a frame to give a whole round to a 512 nodes HSR ring is about 60ms in the worst case; it happens when the longest possible frame is sent by a node, and when this frame has to wait to be resent in each node of the ring to another frame of the same length generated in that intermediate node.

V. NUMBER TRACKING

Discard methods based on number tracking look to sequence numbers received and the order in which they are received to determine if a frame is a duplicate/circulating frame or not.

1) Drop Window

Drop Window is defined in the first edition of the standard to discard duplicates in PRP (it uses different sequence number numeration for each pair source-destination [4], in the second edition each source node uses the same sequence number for every destination, it has reduce the information needed to identify a frame).

The algorithm can be summarized as follows. Nodes maintains per each pair of address source-destination six variables, three per each port (A and B): StarSeqA, CurrentSeqA and ExpectedSeqA; and StarSeqB, CurrentSeqB and ExpectedSeqB.

Operation of the Drop Window method:

- Frames coming from a node are sent in sequence with increasing sequence numbers. Each port maintains for that pair source-destination StartSeq, CurrentSeq and ExpectedSeq. StartSeq indicates the first sequence number of the drop window, CurrentSeq is the just received frame's sequence number and ExpectedSeq indicates the sequence number expected, set to one more than CurrentSeq when a frame arrives.
- If some frames start being received through port A and no frame from B the state of the variables would be as depicted in Figure 6. If then a frame is received through port B it will be dropped or kept depending on the sequence number received, if it is into the drop window of port A it is discarded, if not it is accepted.

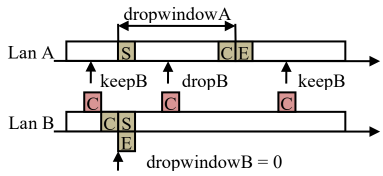
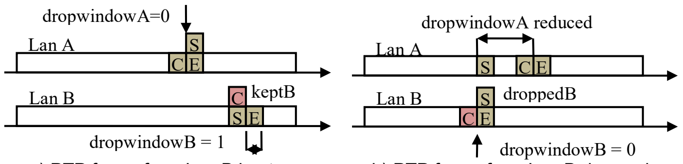


Figure 6. PRP dropwindow on Lan A

- If it is kept, dropwindow of port A is reduced to 0 and dropwindow of port B is set to 1, as sketched in Figure 7a.
- If it is dropped, dropwindow of port B is reduced to 0 and dropwindow of port A is reduced, StartSeqA will take the value of StartSeqB. The result can be seen in Figure 7b.



a) PTP frame from Lan B kept.

b) PTP frame from Lan B dropped.

Figure 7. PTP frames kept and dropped.

Reasons why this method is no more applicable:

- There are cases in which legitimate frames would be discarded: When some frames are received from one port, for example A, and no frames are received from the other, B, and drop window A is increasing as in Figure 6, it supposes that all received frames are correct; but if one or more frames are not received or have error when a following correct frame is received drop window will increase in the same way. Thus if a frame with sequence number into the drop window A is received through port B, it will be supposed to be a duplicate as seen in Figure 7b, although the first copy from port A did not arrive or was wrong, then the opportunity to take the correct frame is lost when it is considered a duplicate.
- Time is not taken into account and each entry should not be remembered for more than 400ms (at 100Mbps).
- This method only works for one duplicate and the second version of the standard states that more than one duplicate can be received.
- Besides, it is not appropriate for HSR because it does not discard circulating frames.

2) Drop Slider Window

Another algorithm, that also uses number tracking, could be based in recording the last sequence number received through each port with a timestamp, for each source. The time mark will be used to delete obsolete entries and estimate how many frames can be considered as duplicate or circulating frames as seen in Figure 8.

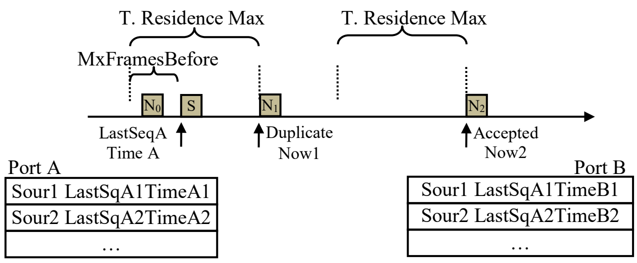


Figure 8. Drop Slider Window. Frames received through port A: N_0 and S. Frames received through port B: N_1 duplicate, N_2 new. If N_1 and N_2 were received through same port A, N_1 would be considered circulating frame.

The operation could be described as follows:

- Each port maintains a list or table with one position per each node in the network to store, node MAC, last sequence number received and timestamp.
- When a frame arrives last time recorded for that source (TimeA or TimeB) will be subtracted to actual time (Now) to obtain time passed from last frame received from that source. If the frame arrives through one port, for duplicates the timestamp used will be of the other port, and for circulating frames the timestamp of the same port will be taken.
- Duplicates:
  - If that subtraction is bigger than Time Residence Max the frame will be accepted and considered new, as depicted in the last frame received in the Figure 8: frame N_2.
  - If that subtraction is smaller; the maximum amount of frames received before the last frame registered can be calculated (MaxFramesBefore) as: Time Residence Max minus previous subtraction divided by the time that takes to the smallest possible frame arriving (70octets at 100Mbps). So that, before the last frame that was stored LastSeqA, as maximum that quantity of frames could have been received from that source. That is to say, if the sequence number of the new frame is in the bracketed interval on (1) the frame is a duplicate. In Figure 8 frame N_1 is a duplicate of N_0.

$$\frac{\mathrm{TResidMxDup} - (Now - TimeA)}{MinimumFrameTime} = MxFramesBefore$$
$$[LastSeqA - MaxFramesBefore, LastSeqA] \qquad (1)$$

- Circulating Frames: The calculus would be analogous, but as said before the time mark of the same port through the new frame has arrived will be considered, and Time Residence Max for circulating frames (this can be the same). If the sequence number of the new frame is in the bracketed interval on (2) the frame is a circulating frame.

$$\frac{\mathrm{TResidMxCir} - (Now - TimeB)}{MinimumFrameTime} = MxFramesBefore$$
$$[LastSeqB - MaxFramesBefore, LastSeqB] \qquad (2)$$

Although this evolved algorithm solves some of the defects of the previous algorithm as time considerations or discarding circulating frames, it has some of its drawbacks. If wrong frames arrive from one port or are lost, and then a correct frame arrives, wrong/lost frames are supposed to be well received, and if the same frame is received after that through the other port, it is considered as received and it is eliminated.

Instead using timestamps other ways to consider time could be used as aging systems shown later or others.

VI. FIFOs (First In First Out) or Circular Buffers

FIFOs or Circular Buffers are another possible way to eliminate duplicates and circulating frames. There are two tables, one per port, in which received frames are stored one after another. When a new frame arrives it must be looked up at the corresponding table, as sketched in Figure 9, to know if that frame has arrived before, and to discard it if it is a duplicate or circulating frame.

If a frame is received through port B it is searched in the table of the port A to see if it has arrived before to determine if it is a duplicate. Moreover, to determine if the same frame is a circulating frame the frame will be looked up in the table of the port B.

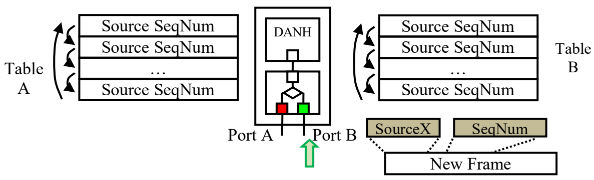


Figure 9. FIFO or Circular Buffer.

Registering arrived frames (Source + SeqNum) has no special requirements; the most restrictive requirements appear when looking up a frame in the memory to confirm if it has arrived before or not, and updating tables. In order to do that memory should be round over in a minimum time.

Reading one position after another by itself could be incompatible with the cut-through mode, used to minimize delay in HSR, in which when addresses and HSR tag are just received, node decides to forward or not; and if it does, forwarding starts before entire frame arrives. It cannot be possible if looking up in memory takes too much time.

So that, FIFOs could be considered as good method to store arrived frames, but a more effective, quicker, access to data should be added. This method could be considered good when number of nodes is not too large, or using an optimal method to look up frames in the tables in a minimum time.

For example, in a 50 nodes HSR ring in the worst case, it takes about 6ms to a frame to give a whole round to it at 100Mbps, in that time a node could, as maximum, receive 6ms/(time to receive the smallest frame, 70octets) =1020.41 frames, so that it is needed at least 1k memory positions. If the number of nodes is increased, number of positions is increased too. With not too many positions a linear search could be used, even if it is not very efficient, or another kind of access could be improved for lecture, as a CAM, which would not be appropriate for large memories [5].

Time considerations must not be forgotten. There can be different possibilities to control it:

- Save for each frame a timestamp and run over the whole memory periodically to ensure that no too old frames are in the table.
- Use an aging system that determines if an entry is too old to be taken into account; for each entry an aging tag is maintained which uses a process running in the background.
- Assure that FIFO is totally run over at a minimum velocity, so that values are rewritten before a concrete time passed, in this case no time marks or tags have to be stored.

Any of those methods or another one which satisfies the time requirements stated in the standard would be valid.

VII. HASH TABLES

As seen in the previous point a main issue using tables is to have an efficient search way, and a way to achieve a quick access to entries in a table is by using hash tables.

The idea of hash tables is to apply a hash function to source address + sequence number; the result will be used as index to find the place in the table for that entry. By looking in that place it will be seen if the frame has arrived before, and there it will be stored the frame that has arrived, as seen in Figure 10. There are different algorithms to implement a hash table and resolve the collisions (when different source + sequence number have same hash).

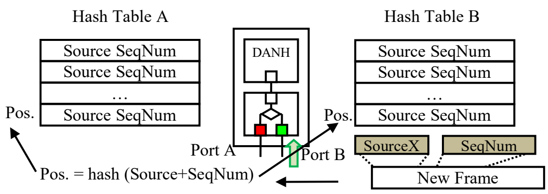


Figure 10. Frame management using hash tables.

The use to detect duplicates and circulating frames is the same as in the previous tables; the difference is the way to index the tables.

Hash tables and circular buffers were analyzed in [6] for HSR/PRP version 0. Circular buffers resulted by itself having good performance of rejecting duplicated frames but having the longest searching time to make the decision (drawback for HSR due to time requirements). But it obtained good results with a combined method (hash tables plus circular buffers) for networks with less than 64 nodes.

Tables must have a parallel time control using one of the methods seen before or another one which satisfies the standard specifications.

VIII. NODE TABLES

A node table is a table where every node of the network is registered. Those tables were obligatory in the previous version of the standard [4], but new version [2] makes them optional, so that they would be used on those nodes from which the network will be managed. The state of the network will be stored in the table using variables as number of frames received from each port, number of frames with error or time at which the latest frame was received; basic data is enumerated in the standard.

There are two possibilities with node tables (in the nodes they are implemented):

- Use one of the methods seen before or another one to discard duplicates and circulating frames and a parallel node table to store information about the state of the network. This option could be the most eligible for hardware implementations in which there will be hard time requirements.
- Use node tables, which store more information, in order to eliminate duplicates and circulating frames too. This case would be appropriate for software versions and configurations with more relaxed time requirements.

Node tables will be indexed by MAC address so that every time a frame arrives, the table will be revised to see if MAC address is in the list and then, with the sequence number, if that frame has arrived before. Obviously this solution should need an optimized search for those applications in which time is important and cut-through is implemented. Moreover, as said before tables must have a time control to eliminate obsolete information.

IX. Reduction to One Table

Every seen method maintains one register or table per port were source address + sequence number of received frames are saved. When a frame arrives, it is looked up in each table to see if it arrived before to see if it is a duplicate, circulating frame or new. This supposes to maintain at least two tables for nodes and three for RedBoxes (3 ports), so it should be extremely useful to reduce the number of tables to only one in which search each new frame.

By saving in a table, with source + sequence number, some bits which say from where the frame has arrived it could be done. Then when a new frame arrives it is looked up in the table and if it is found it must be decided if it has to be forwarded or sent to upper layers or neither. Figure 11 shows a node with a unique table in which arrived frames are stored and port from which they have arrived before.

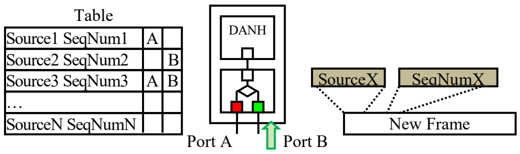


Figure 11. Frames management with a unique table.

In parallel when (time) the frame was received must be considered using timestamps, aging, rewriting old entries or other method.

In case of RedBoxes, for example when a PRP an HSR networks are interconnected, frames arrived through three ports must be controlled. As depicted in Figure 12 a frame generated from the PRP network from one DAN, 1, generates two frames, 1A and 1B, and the interfaces with HSR network, RedBoxes, generate two new frames. But it must be heeded that a RedBox which receives a frame generated in the other RedBox only has to forward one frame in each direction. So that, RedBox1 should send in anticlockwise direction 1AA or 1BA, depending on which one arrives before, 1A from the interlink or 1BA from the RedBox2. For RedBox management the use of a single table is easier than three.

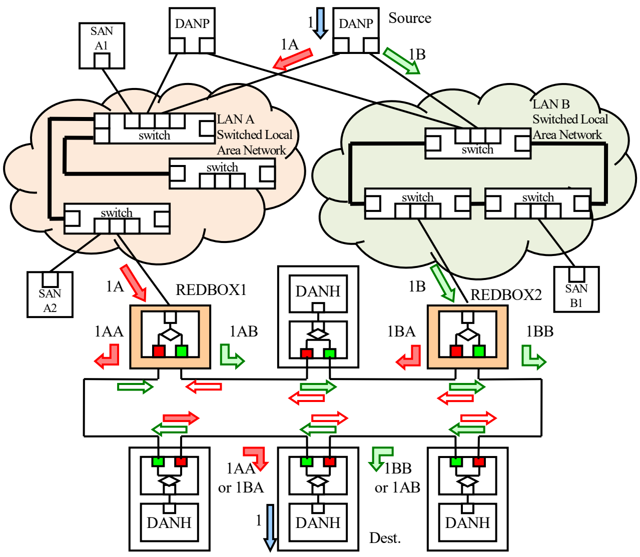


Figure 12. PRP an HSR interconnection by two RedBoxes.

Another variation of this method which could be very useful could be to remember through which port the frame has been sent, instead of using the receiving port.

X. INCORRECT FRAMES

Finally, there must be taken into account when a frame is incorrect to mark it, or when it is marked as erroneous with the error sequence as the standard states. Even if the frame is erroneous it must be registered in the table in case it is received again, as circulating frame, to discard it; or in case the frame was received with error through one port and it can be received correctly through the other one.

So apart from a bit per port to indicate through which port the frame has arrived/has been sent there should be another bit per port to mark if the frame was erroneous in that direction.

XI. CONCLUSIONS AND FUTURE WORK

Summing up, the analysis of the duplicate and circulating frames discard methods for PRP and HSR shows that:

Number tracking studied methods are not appropriate by themselves; they could be used for some situations if they are combined with other methods, or using another different algorithm that solves the analyzed weaknesses.

Methods based on circular buffers (FIFOs), hash tables and node tables solve the drawbacks of the previous based in number tracking, but have other challenges.

Circular buffers or FIFOs could be a good system in which new entries are easily saved but has the drawback of the need of an optimal way to search entries in a minimum time. The problem is increased by increasing the number of nodes and traffic.

Hash tables, as the other table based methods, consist on maintaining tables with information about frames arrived through each port. The improvement of hash tables is the rapid access to positions in tables but it has the problem of collisions.

Node tables, mandatory in the previous version of the standard, permit each node to have a record of the status of the whole network. The new version considers them optional, so that they could be implemented only in nodes from which network would be managed. Discarding method could be one of the previously seen or other, having a node table in parallel either in software or in hardware to monitor the health of the network.

Some improvements for discarding tables have been proposed: The reduction of the amount of tables to one so as to simplify the system and memory requirements considerably, and having error frames into account properly.

In order to optimize tables and simplify discarding methods it would be very useful to analyze the requirements and characteristics of the networks and traffic used by the particular applications. For example, the draft of the IEC61850-90-4 presents PRP and HSR as redundancy protocols for high availability in substations networks, for Station Bus and Process Bus with its characteristic traffic type [3].

The developed study will be very useful for the implementation of the duplicate and circulating frames elimination method in a real system, to develop nodes and RedBoxes for PRP and HSR networks.

ACKNOWLEDGMENT

This work has been carried out inside the 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 research program SAIOTEK (project S-PE12UN008).

REFERENCES

[1] *IEC 62439-1 Ed.01: Industrial Automation Networks - Part1: General concepsts and calculation methods,* IEC, 2010.

[2] *IEC 62439-3 Ed.02: Industrial Communication Networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and Hig-availability Seamless Redundancy (HSR),* IEC, 2012.

[3] *IEC 61850-90-4 (draft): Communication Networks and Systems in Substations - Part 90-4: Network Engineering Guidelines.,* Geneva: IEC, 2012.

[4] *IEC 62439-3 Ed.01: Industrial Communication Networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and Hig-availability Seamless Redundancy (HSR),* IEC, 2010.

[5] K. L. (Xilinx), "XAPP1151 (v1.0) (Parameterizable Content-Addressable - Application Note: Xilinx FPGAs)," Xilinx, http://www.xilinx.com/support/documentation/application_notes/xapp1151_Param_CAM.pdf, 2011.

[6] J. Xiaozhuo, High Availability Seamless Ring Protocol Inplementation in FPGA (Master Thesis), 2009.