Formal verification of redundant media extension of Ethernet PowerLink

Steve Limal, Stéphane Potier, Bruno Denis, Jean-Jacques Lesage

Formal verification of redundant media extension of Ethernet PowerLink

Steve LIMAL, Stéphane POTIER
Alstom Power - Power Control Systems,
9, rue Ampère,
F-91345 Massy Cedex, France
{limal, potier}@power.alstom.com

Bruno DENIS, Jean-Jacques LESAGE
LURPA, ENS Cachan, UniverSud Paris
61, av. du Président Wilson,
F-94230 Cachan, France
{denis, lesage}@lurpa.ens-cachan.fr

Abstract

The use of Ethernet at the field level seems to be the next step after traditional fieldbusses. Even if it was not used to be competitive compared to solutions designed for industrial purpose, Ethernet performances have increased faster. On the other hand, some special features like network availability solutions have not improved so much rapidly. Then faster Ethernet based industrial protocols had to specify accurate solutions.

The objective of this paper is to validate the medium redundancy management part of the Ethernet PowerLink High Availability extension. For this, aimed application requirements are stated and the protocol with its extension are detailed. In the context of Alstom Power critical applications, the correctness of the solution of redundancy must be proven. Therefore, a model-checking approach is used from a generic modelling in timed finite state automata.

1. Introduction

Automation technology requirements increase continuously both in terms of amount of process data or supervision data or in terms of process data refreshment rate. As a result, current fieldbusses like WorldFIP or Profibus fall short of performances when having to apply to some today's applications. For example, if power production processes have not drastically changed for years, amounts of inputs and outputs have grown while field network cycle times have decreased. The reason is that control must be more and more accurate when more process efficiency is aimed. Furthermore, as said in [9], throughput-oriented communications have well grown. Alstom Power[1] has followed the incoming of new Ethernet-based fast network protocols which might be used at the field level when WorldFIP performances no more fit.

Due to intrinsic indeterminism of CSMA-CD (for Carrier Sense Multiple Access with Collision Detection) Ethernet's access to medium, academic work followed different tracks to apply it to industrial applications. New means of medium access have been proposed, whether by defining new Medium Access Control layer, like in [15], or by going round of currently used IP and TCP layers like in [11]. Evaluation of achievable performances on IP (Internet Protocol) based industrial protocols or new Ethernet based fast network protocols have also been made like in [12] or [6]. As regards network availability, works like in [18] from the NECST project mainly deals with IP solutions and do not consider faster Ethernet based protocols. As these are quite new, no work has been done yet to evaluate their availability solution. Thus as availability are designed for critical applications like Alstom power plants, every doubts on achieved properties must be revoked.

Among precautions and procedures currently taken to develop a dependable system, formal verification has been chosen to validate the availability feature of a network architecture with a "hard real-time" Ethernet based protocol. Indeed, considered controlled systems are critical and the use of real-time Ethernet at the field level is quite at its beginning in this context. Then formal verification enabled to take care and insure agreement with power generation sector concerns since conception. In the following part, the aimed application requirements will be given before explaining the focused Ethernet based protocol and its network availability principles. Then the chosen verification technique and the modelling will be explained. Finally, a chosen category of architecture will be treated and the results will be given and commented.

2. Application requirements

Figure 1 illustrates a typical architecture of an Alstom Power control system. This architecture is applied in many types of power plant. It is composed of automation cells connected on a supervision network. The supervision network enables several automation cells to exchange data with each other and supervision stations. This network is currently already supplied by a dual-ring Ethernet network as required real-time level is lower. At the field level, each automation cell hosts a distributed automation network and assume one or more process functions through up to 3000 inputs and outputs. This paper

[1] Alstom Power finances first author study grant with a program from the French Association Nationale de la Recherche Technique (ANRT).

focuses distributed automation network. A cell network function is to assume data transmission between the following nodes: A redundant cell controller and a score of field controllers and remote I/O stations. Thereafter, the word “network” will stand for the set of hardware and software components which participate to this function of data transmission, i.e., the interconnection devices and cables, the coupling hardware of the nodes and the protocols involved.

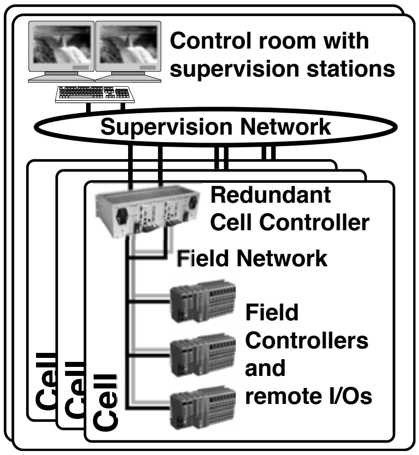
Control room with supervision stations
Supervision Network
Redundant Cell Controller
Field Network
Field Controllers and remote I/Os
Cell
Cell
Cell

Figure 1. Control system architecture overview.

In order to apply the generic concept of cell to many power generation processes (for hydro, steam or gas turbine, but also boiler control functions), the following wide-ranging features and achieved performances requirements must be encompassed by the cell network:

- behavioural determinism. The common system requirement is to obtain temporal consistency of data by means of isochronous data transmission.
- High availability. Failure of one component of the network must not lead to discontinuation of the control on the process.
- Real-time class 2 (according to [14]) reactivity. A network cycle time of 5ms for up to 25 nodes sharing up to 7kB of process data must be achievable.
- Free Topology. The network architecture must be flexible as it can result in a tree topology extended on kilometers in power plants like China’s Three Gorges Dam. The ability to mix copper and optic fiber media is also necessary. It implies that topology cannot be optimized like proposed in [7].

Safety improvement by mean of management of the data integrity is not a requirement. Indeed, we share the “black channel” approach of IEC61508. The network is seen as unreliable and an independent safety communication layer sharing the concerns of IEC 61784-3 (Profiles for functional safety fieldbus) must be used.

Among solutions soon standardized in IEC61784-2 and introduced in [5], solutions which rely on IP, without IEEE1588 synchronization, cannot guarantee data isochronism for the required real-time level. In fact, only Profinet IRT, Ethernet PowerLink, EtherNET-IP+CIPsync, EtherCAT and SERCOS III look capable enough regarding to requirements. Given the fact that Ethernet PowerLink communication model well suits to distributed automation and enables a good topology flexibility, Alstom Power Control Systems has chosen to first integrate it in its ALSPA P320 Distributed Control System solution in 2005 (all the more, it was more developed than others solutions at this date).

3. Ethernet PowerLink technology

In 2002, Bernecker&Rainer automation devices manufacturer opened its Ethernet PowerLink (EPL) protocol to third party use. Since 2003, EPL is managed by the Ethernet Powerlink Standardization Group, independent association bringing together manufacturers and users of the EPL protocol.

3.1. Ethernet PowerLink (EPL) protocol

EPL is based on standard IEEE802.3 layers. This means that EPL stack can be implemented whether by hardware or by software with a standard Ethernet chip. Upper layers are EPL partly specific up to devices profile layer which reuse those specified by CANopen for process data communications. Figure (2) shows where EPL layers insert in the OSI model respect to the darker most known layers.

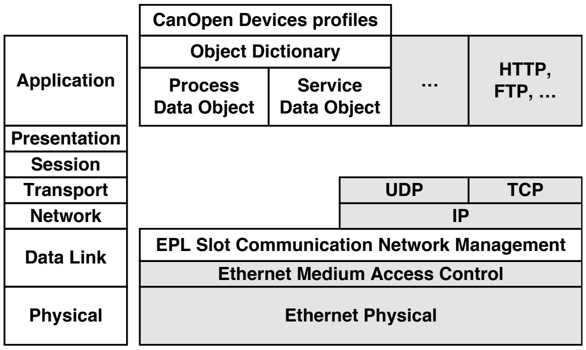

Figure 2. EPL layers respect to OSI 7498 model.

In order to achieve temporal performances and behavioural determinism, an EPL network must be isolated on a sub-network. This enables nodes exchanges to be supervised by a network arbiter called Managing Node (MN). The other nodes, called Controlled Nodes (CN), won’t send frame on the network until invited by the Managing Node. EPL follows the publisher-subscriber “pull-model” reminded in [17] and defined in IEC61158.

The use of this relationship model enables not triggering CSMA-CD Ethernet mechanism then enables determinism even on a shared (i.e., only using repeating hubs) network. Segmentation in sub-networks dedicated to EPL, more than representing an issue, well suits to the control system architecture illustrated in figure 1.

An EPL elementary cycle is composed of three main phases as shown in figure 3 with taking into account one MN and two CN:

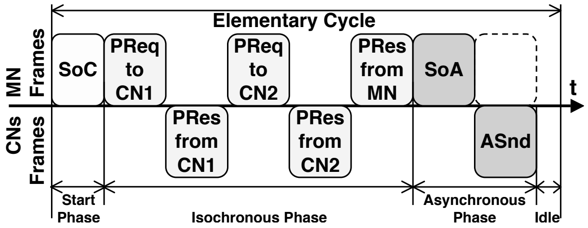

Figure 3. Example of EPL elementary cycle with 1 Managing Node and 2 Controlled Nodes.

- Start of Cycle (SoC) phase. This short phase enables to indicate the start of cycle to all the nodes and to synchronize nodes actions with only one multicast frame sent by the MN.

- Isochronous phase. During this phase, the MN will poll each CN with a Poll Request (PReq) unicast frame containing Process Data Objects (PDO) and allowing CN to multicast their own PDO in their Poll Response (PRes) and ask for right to send asynchronous frame. MN can also multicast a PRes without requesting itself. This phase is configured so that exchanged PDO are always the same.

- Asynchronous phase. During the asynchronous phase, the MN elects and invites itself or one CN to send an asynchronous frame in the Start of Asynchronous (SoA) frame. The Asynchronous Send (ASnd) frame can be standard IP frame and will be used for Services Data Objects (SDO) commands between nodes like configuration or remote access. An idle time can occur at the end of the cycle depending on configured cycle time and effective length of the previous phases.

Neither the EPLsafety extension nor functional safety communication profiles of IEC61784-3 is considered here. In the following section, we will focus the EPL High Availability extension which is to be approved (in spring 2007). This extension specifies achievement of better availability with EPL.

3.2. High Availability extension

In December 2005, some EPSG members decided to work together to specify an High Availability add-on to the EPL Communication Profile Specification. The purpose was to enable not stopping process control in case of failure of any component. They studied possible solutions which can offer the following properties in response of their applications requirements (including those described sooner):

- Tolerance to one medium failure is necessary. Network must continue to run even if one medium component fails, if a cable is broken or an interconnection device breaks down.

- Tolerance to any node failure (or only its coupling board break down) is necessary if the function to grant access to the medium is centralized in a unique arbiter node.

- The fewest frame loss must be achieved if a failure occurs. This gives the responsiveness of the redundancy solution.

- Any failure must be detected and reportable up to supervision in order to trigger maintenance action.

The resulting EPL High Availability extension specification [4] deals with:

- Protocol Redundancy achieved through Managing Node redundancy. Arbiter redundancy must be achieved, when using the publisher-subscriber “pull-model”, to insure transport producing in case of arbiter failure. Thus communications dependability is increased.

- Redundancy of hardware architecture. As communications must still be done even in case of a medium component failure, data transport availability is also improved.

- Compatibility. As the “extension” term means, EPL High Availability is not a different communication protocol but is designed to allow the use of existing Controlled Nodes.

Depending on application requirements, whether only protocol redundancy or redundancy of hardware architecture will already increase dependability. For the most critical applications, the two will be necessary. in the following, taking into account of paper length, we will develop only the redundancy of hardware architecture part.

4. Design of redundancy of hardware architecture

Together with Ethernet evolutions, office Information Technology has brought network availability solutions whose healing time in case of failure is adapted to office needs. But if we consider path redundancy protocols like Rapid Spanning Tree Protocol (IEEE 802.1w) or Link Aggregation (IEEE 802.3ad), their best reconfiguration time cannot be less than 1s. On the other hand, less compelling applications can need a network cycle time of some hundreds of milliseconds. Some manufacturers have proposed proprietary solutions, mainly based on rings as described in [8]. But they do no suit with fast applications using EPL as ring solutions healing can take from 20 ms (for the best ones) up to 500 ms. This means that with 5ms cycle time, hardware architecture can take from 4 cycles to 100 cycles to self repair. But path redundancy is not only targeted, healing time must be short and adapted to the network cycle time.

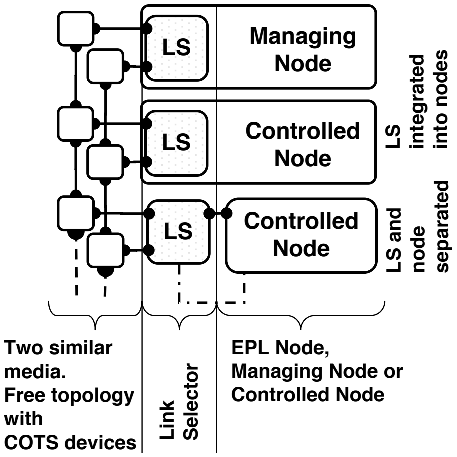

Figure 4. Studied architecture overview.

The retained solution (illustrated in figure 4) relies on:

- Two independent media. The linked constraint is that the two medium layouts must be the same in terms of interconnection devices hops between two nodes. On the other hand, this solution does not impose a new network redundancy protocol embedded in each interconnection device. Then any existing Commercial Of The Shelf (COTS) devices can be used. Therefore, only broadcasting components (cable, hubs, media converters) will be considered here.
- Medium redundancy management. It is performed at the nodes level. An Ethernet Link Selector (LS) provides this management of the medium redundancy, fault detection, recovery strategy and eventually notification of the fault. This function can be performed internally into the node or by an additional device serving the node.

As a result, the Link Selector apply the “Workby redundancy” defined in IEC61078 and reminded in [10]. It enables a node to have a link on each of the independent media. Its principle is illustrated in figure 5 and can be summarized in few points :

- any frame from a node will be duplicated and sent on each medium.
- the redundant frames will be received in parallel by each LS serving nodes. Thus LS must be tolerant to delay arrival between the two frames.
- LS will only forward one of the two received frame to the upper layers or to the single port of the served Node. Frame selection criteria are not imposed.

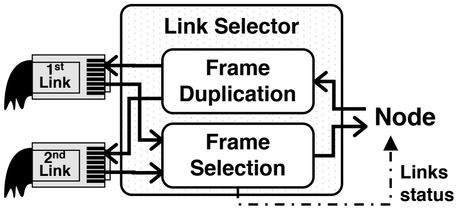

Figure 5. Link Selector performed functions.

It must be noticed that fault detection and recovery strategy is designed to be fully independent of the EPL protocol as soon as communications are deterministic. Indeed frames have to arrive in the same order on the two links of a node not to trigger frame loss errors. Fault notification up to supervision is a bit harder to free from protocol if the Link Selector is running outside served node, but adding connection between it and external inputs of the node.

Since only half duplex is necessary for its communications, the EPL protocol advices but does not comply the use of hubs instead of switches. Even if said to be surmounted by switches, hubs are five times cheaper than unmanageable switches. Furthermore, a hub introduces five times to more than hundred times lower delay than a Fast Ethernet Store-and-Forward (S&F) switch (S&F switch type is the only available on the market). Finally, as a hub is simpler, we can expect it to be more reliable. That’s why switches are not considered by this paper.

The principles of the EPL architecture redundancy described in this part, even if thought to be simple have to be implemented in a short time and will be used in critical systems. Then some doubts have to be lifted in case of medium failure, such as:

Q1 Can redundancy miss a failure?

Q2 Can some frames be lost by upper layers?

Q3 Can selected design avoid triggering CSMA-CD?

Q4 What is components delay and jitter influence?

Choice have been made to answer these questions by mean of formal verification. By exploring all possible states reached by the redundant architecture, it has been possible to validate the designed solution before its implementation. The next parts describe our modelling and the formal verification process.

5. Formal validation of the designed solution

In order to make sure that medium redundancy principles cannot bring an application in an unwanted state, choice has been made to apply formal verification on a modelling of behaviours.

5.1. Model-Checking principles

Formal verification techniques stem from the field of computer science. Only recently they have been adapted and applied to Discrete Event Systems (DES) verification [13]. The most common techniques used on this field are the theorem proving [16] and the model-checking [2]. The general principle behind model-checking may be expressed as follows (see Fig. 6). Let's start with a system that has been designed to verify an entire array of properties (logical correctness, dependability, liveness, etc.). The first task consists of formalizing system behaviour in the form of a finite state automaton: S, plus the properties to be verified within a temporal algebra such as Computation Tree Logic (CTL) [3]: φ. The model-checker then conducts a thorough analysis of the state space reachable by S, which serves either to prove that $S \models \varphi$ (this algebraic statement denotes that "the system model satisfies the set of properties φ") or, when such is not the case, to propose a counterexample that revokes those properties not verified by S.

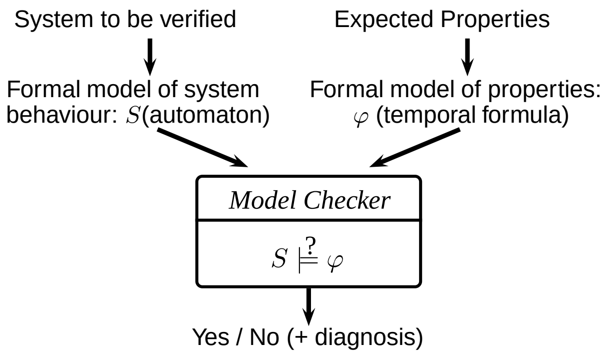
System to be verified
Expected Properties
Formal model of system behaviour: S(automaton)
Formal model of properties: φ (temporal formula)
Model Checker
$S \overset{?}{\models} \varphi$
Yes / No (+ diagnosis)

Figure 6. Model-checking scheme.

The modelled system is a cell made of devices whose individual behaviour and interactions are time dependent. Indeed individual change of state will often be triggered by reception of messages whose propagation time cannot be considered as negligible with considered real-time level. Thus temporal model-checking was chosen. The selected timed model-checker tool to preform modelling, validation and verification of the cell is UPPAAL [1]. This tool has already been applied to industrial real-time systems and is regularly improved.

5.2. Generic and modular approach

UPPAAL enables templates of timed automata to be instantiated in order to describe a final set of timed automaton. Then a cell (as described in the application requirements section) hosting an EPL network with any redundant hardware architecture can be described by instantiation in a list of generic behaviours. The figure 7 gives a structural view of a cell network with EPL modelling.

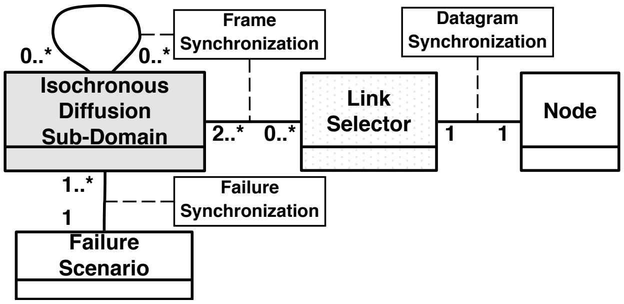

Figure 7. UML class diagram of the EPL network modelling.

Every model of EPL network is described from instances of templates modelling four types of behaviour; the behaviour of a component (a Node or a Link Selector), the behaviour of a group of components (forming an "Isochronous Diffusion Sub-domain") or the behaviour of a supervisor triggering a failure on the redundant medium according to a defined scenario.

The modelling has been made so that there is no need to adapt one of the following template: (The timed finite state automata models of figures 8, 9, 11 and 12 are made "readable" for this paper. UPPAAL states type "Urgent" and "Commit" are removed but state type "Initial" is maintained. Transition labels are simplified so that they can be action or guard depending on label semantic (action verb or event). Moreover, neither template internal variables nor time management are detailed.)

Node. This template (figure 8) models the behaviour of a node running normally, just after startup. First, a node model will switch in Managing Node behaviour (right) or in Controlled Node behaviour (left) depending on the parameterization of each instance. Then it will manage the exchange of "datagram synchronization" events (transitions labeled with Send verb or received event) with dedicated Link Selector and the change of state according to the cycle illustrated in figure 3. As this modelling purpose is to study only failures in the architecture, no node failure is triggered but model will enter a sink CNerror state on error detection. Parameters for this template are the node identifier, the time features (describing device design like delay from request reception to response sending) and the time parameters (depending on device setup).

Link Selector (LS). This template (figure 9) models the principles of frame duplication and frame selection illustrated in figure 5. From Inactive initial state, when a "datagram synch." is received from dedicated node (transition labeled Datagram received from node), the LS model broadcasts a "frame synchronization" event (transition labeled Send frame on the two ports) towards each of the two media (cf. figures 5 and 7). From its Inactive initial

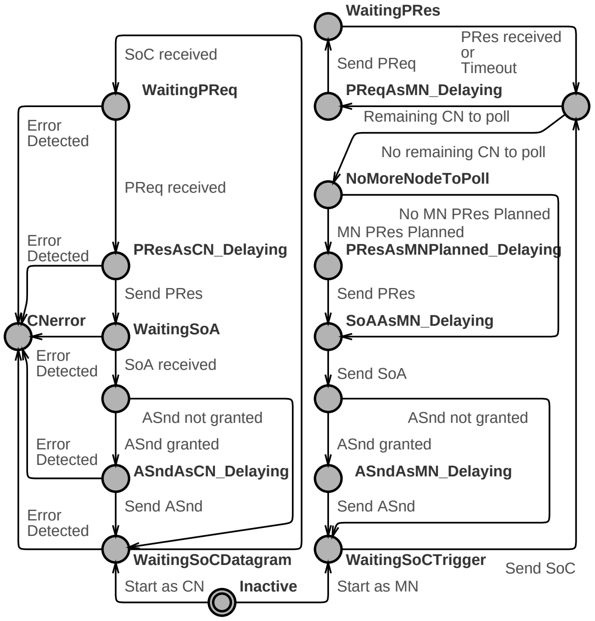

Figure 8. Simplified Node template.

state, when receiving a "frame synch." (transition labeled Frame received)) from the media, the simplest possible strategy of frame selection is modelled. First received "frame synch." parameters from a medium are selected and forwarded to the node via a "datagram synch." (transition labeled Send datagram) as soon as redundant "frame synch." is received or after a timeout. During selection, any "frame synch." from the same medium is treated as an error leading to the sink LSerror state. The timeout is constant and is equal to the minimum time, e.g., $5.7\mu s$ for Fast-Ethernet. Parameter for this template is the served node identifier.

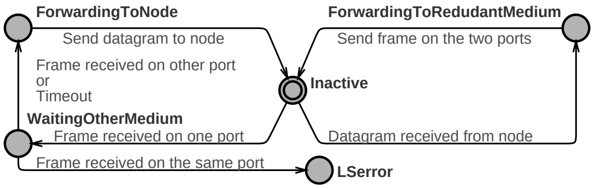
ForwardingToNode
ForwardingToRedudantMedium
Send datagram to node
Send frame on the two ports
Frame received on other port
or
Timeout
Inactive
WaitingOtherMedium
Frame received on one port
Datagram received from node
Frame received on the same port
LSerror

Figure 9. Simplified Link Selector template.

Isochronous Diffusion Sub-domain (IDS). As already said, the modelling only considers a shared medium. As a result, every component on each medium will be in the same diffusion domain. In order to model a medium, we split its diffusion domain into one or more parts we call Isochronous Diffusion Sub-domains (IDS). From Inactive initial state (figure 11), an IDS has the property to forward a received "frame synch." from an edge (transition labeled Frame received) to the others edges after a variable delay (transition labeled Send frame). The edge term refer to a frame communication link between automata (IDS or LS). The delay and its variation, called jitter, will depend on the number and type (Hubs, cables, media converters...) of components which are encompassed by the IDS.

As illustrated in figure 10, when a "frame synch." is received at an edge E_i of an IDS whose parameters are the delay ($Delay > 0$) and the jitter ($Delay > jitter \geqslant 0$), this synchronization will be forwarded to any LS or IDS plugged at an edge E_j after a time $Time(E_i, E_j)$. $Time(E_i, E_j)$ is defined by the relation: $\forall\ i, \forall\ j \neq i, Time(E_i, E_j) = Delay + \triangle t$ with $|\triangle t| \leqslant \frac{jitter}{2}$.

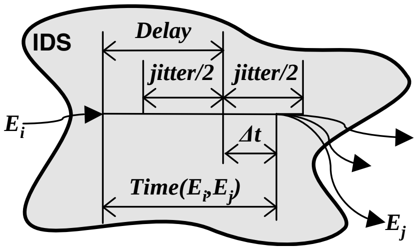

Figure 10. Isochronous Diffusion Sub-Domain (IDS).

As shown in 11, from Inactive initial state, an IDS can reach a failure state where synchronization propagation will not be performed any more. After 2 to 3 cycles in failure state, IDS will self repair and restart forwarding received synchronizations. From BusyDelaying state, automaton will go into a sink IDSerror state if it receives another "frame synch." (for collision detection).

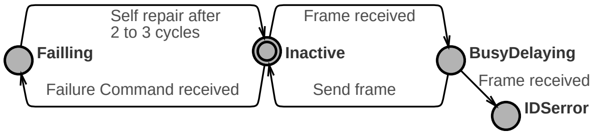

Figure 11. Simplified IDS template.

Failure Management Scenario. This template (figure 12) enables to command instantiated automata to enter a failure state. In our case, a single scenario model will command all failures. After one network cycle without any failure triggered (considered as normal running), our failure management scenario will command failure of one of the IDS before the end of the current network cycle.

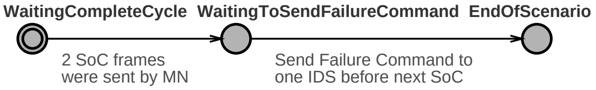

Figure 12. Simplified Failure Scenario.

Exchanged data are not modelled. Only their transmissions are modelled by the use of event messages. Triggering of failures are also modelled with event messages.

In order to make the links (i.e., the shared "frame synch.") between all instantiated automata, a matrix of boolean tells each instance of IDS or Link Selector model with which other models they synchronize. The Datagram synchronization event between nodes and Link Selector automata is shared according to the Node ID parameter.

In the next part, the questions raised in previous section are traduced and verified on a category of architecture.

5.3. Verification of a category of architecture

This section studies a cell hosting a category of redundant hardware architecture. As illustrated in the example of figure 13, it is constituted of a double line topology where each node connect to the redundant "backbone" medium thanks to repeating hubs. For the modelling, each diffusion domain has been split into Isochronous Diffusion Sub-domain so that each repeating hub with part of connected cables is an IDS. We set every IDS with the same couple of parameters $(Delay, jitter) = (500ns, 50ns)$ (figures taken from manufacturers documents). The figure 13 gives an example of how a cell network with the considered category of architecture is modelled. For a cell hosting 3 nodes (on the left), the UPPAAL model of the cell has been instanciated according to the corresponding UML object diagram (on the right). The object diagram is derivated from the class diagram of figure 7 .

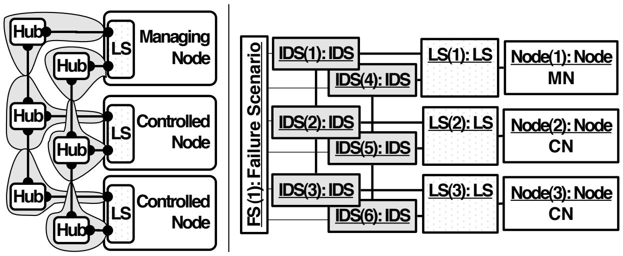

Figure 13. Modelling example with 3 nodes.

The verification has been applied on a PIV 2.4GHz computer under GNU-Linux 32bits with UPPAAL 4.0.6. The next table summarizes the number of UPPAAL clocks involved, the number of stored states, the maximum of physical memory used and the time to compute the verification for a growing number of nodes.

Nodes	Clocks	States	Memory	Verif. time
3	15	57570	16MB	30s
4	20	331613	86MB	8min
5	25	1355167	380MB	1h15min
6	30	5514884	1,7GB	14h30min

With more than 6 nodes, too many clocks are involved in the model, and the state space to be explored is too wide when verifying the properties given hereafter.

The following properties (expressed in a simplified version of CTL) enabled to check the EPL extension design correctness (PA), the particular architecture parameterization correctness (PB) and the model constistency (PC):

-
  ```
  (IDSFailures>=1) --> (exists
  (i:  int[1, NbNodes])
  LS(i).LateMediumCounter>0)
  ```

  PA: If a medium failure occurs, at least one Link Selector must detect a late frame error and increase its LateMediumCounter counter (necessary to answer Q1 in section 4).

-
  ```
  A[] not (exists (i:  int[1, NbIDS])
  IDS(i).IDSerror==1)
  ```

  PA: No collision will occur on the network due to the specification of the redundancy behaviour (Q3).

  PC: No message is lost because an IDS is receiving a "frame synch." event while delaying previous one.

-
  ```
  A[] not (exists (i:  int[1, NbNodes])
  Node(i).CNerror ==1)
  ```

  PA: Controlled Nodes do not fall in CNerror sink state because no SoC has been received for too long time or SoC, PReq or SoA frame has been lost (Q1, Q2, Q4).

  PB: CN has the correct parameterization to be tolerant to a medium failure (Q4).

-
  ```
  A[] not (exists (i:  int[1, NbNodes])
  LS(i).LSerror==1)
  ```

  PB: LS won't receive a frame on a port while waiting for a redundant frame on the other port (Q4).

-
  ```
  A[] not deadlock
  ```

  PC: There is not any deadlock. The model consistency should be checked first, but analyst cannot conclude on it without previous properties. Indeed PA and PB issues cause deadlocks too because of the used sink states.

The different properties verification enabled to confirm expected properties as well as expected limits of the modelling :

- In case of only one medium failure, failure is detected (Q1), no frame is lost (Q2) and no collision occurs (Q3). As a result, the protocol redundancy will be able to refer to frame reception in order to recover rapidly.

- Maximum delay of an IDS model must be inferior to any node's model delay between two consecutive frame synchronization sends. Otherwise, the IDS model will consider a collision has occurred and will enter in a sink state. Then exceed of this limit prevent the verification of such rejected behaviour (Q4).

- The minimum frame selection principle which is modelled will erroneously detect medium failure if the difference between, in one side, the sum of all minimum delays on a medium, and on the other side, the sum of all maximum delays on the other medium, is greater than the $5.7\mu s$ timeout value (Q4). The advantage of the model-checking technique is to detect these cases as all paths between nodes will be covered.

If the aimed maximum number of nodes has not been reached, the parameters limits which have to be observed are well identified. Indeed, the obtained results apply to any size of network. Moreover, the purpose was to verify the defined category of architecture and the split into IDS has been thought only to be generic. The split could be optimized to reduced the number of IDS. Then nodes could be added thanks to the saved IDS'clocks.

6. Conclusion and future work

In order to achieve a solution offering both availability and compliance with the performances of the EPL protocol, the specified redundancy of hardware architecture relies firstly on two independent but similar media (with COTS devices) and secondly on the function Link Selector performed at the node level. It has been necessary to make sure that these two principles would provide the required availability properties. And temporal model-checking has behaved as a technique of choice to validate the sooner the main options. The verification of timed models enabled not only to validate the expected logic properties as non-temporal model-checking can do, but it also enabled to determine temporal domains where properties are observed. As the delays introduced by the medium components in communications cannot be ignored, we were able to conclude with a typical, delayed, hardware architecture.

The readily adjustable modelling of the redundant medium into Isochronous Diffusion Sub-domains allow the abstraction level to be adapted (in order to make the verification computable). For practical purpose, thanks to the generic and modular approach, temporal model-checking together with a strong abstraction of the redundant medium enable to verify a project of EPL network with constraints. These constraints are first a particular topology of the hardware architecture with many nodes and secondly the components offer. Then temporal model checking becomes a tool to help configuration respect to the properties of dependability.

The future work is to model the Managing Node redundancy by enhancement of the node template. This will enable to check if only one MN can be active on a single hardware architecture free of failure. After independent validation of the redundancy of hardware architecture and the protocol redundancy, the next step will be to verify if the two can work together without any loose of property. For example, a minimum recovery time must be achieved and failure from a medium must not wrongly trigger the protocol redundancy.

References

[1] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi. UPPAAL — a Tool Suite for Automatic Verification of Real-Time Systems. In *Proc. of Workshop on Verification and Control of Hybrid Systems*, pages 232–243, 1995.

[2] B. Bérard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, and Ph. Schnoebelen. *Systems and Software Verification. Model-Checking Techniques and Tools*. Springer, 2001.

[3] E. Emerson and J. Halpern. “Sometimes” and “not never” revisited: on branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.

[4] EPSG. *Ethernet PowerLink V2.0 High Availability Specification. Working Standard Proposal, Version 0.1.3*, 2007.

[5] M. Felser and T. Sauter. Standardization of industrial Ethernet-the next battlefield? In *Proc. of IEEE Workshop on Factory Communication Systems*, pages 413–420, 2004.

[6] P. Ferrari, A. Flammini, and S. Vitturi. Performance analysis of PROFINET networks. *Computer Standards & Interfaces*, 28(4):369–385, 2006.

[7] J.-P. Georges, N. Krommenacker, T. Divoux, and E. Rondeau. A design process of switched Ethernet architectures according to real-time application constraints. *Engineering Applications of Artificial Intelligence*, 19(3):335–344, 2006.

[8] K. Hansen. Redundancy Ethernet in Industrial Automation. In *Proc. of 10th IEEE Conf. on Emerging Technologies and Factory Automation*, number 7, 2005.

[9] J. Jasperneite and P. Neumann. How to Guarantee Realtime Behavior Using Ethernet. In *Proc. of 11th IFAC Symp. on Information Control Problems in Manufacturing*, pages 115–140, 2004.

[10] H. Kirrmann and D. Dzung. Selecting a standard redundancy method for highly available industrial networks. In *Proc. of 6th IEEE Workshop on Factory Communication Systems*, pages 386–390, 2006.

[11] J. Kiszka, B. Wagner, Y. Zhang, and J. Broenink. RTnet — A Flexible Hard Real-Time Networking Framework. In *Proc. of 10th IEEE Conf. on Emerging Technologies and Factory Automation*, 2005.

[12] G. Marsal. *Evaluation of time performances of Ethernet-Based Automation Systems by simulation of High-level Petri Nets*. PhD thesis, ENS Cachan (France) and Univ. of Kaiserslautern (Germany), 2006.

[13] I. Moon. Modeling programmable logic controllers for logic verification. *Control Systems Magazine, IEEE*, 14(2):53–59, 1994.

[14] P. Neumann. Communication in industrial automation–what is going on? *Control Engineering Practice*, (2006), doi:10.106/j.conengprac.2006.10.004.

[15] S. Ouni and F. Kamoun. Hard and soft real time message scheduling on Ethernet networks. In *Proc. of IEEE Conf. on Systems, Man and Cybernetics*, 2002.

[16] J. Roussel and B. Denis. Safety properties verification of Ladder Diagram programs. *Journal européen des systèmes automatisés*, 36(7):905–917, 2002.

[17] J.-P. Thomesse. Fieldbus technology in industrial automation. *Proceedings of the IEEE*, 93(6):1073–1101, 2005.

[18] N. Vatanski, J.-P. Georges, C. Aubrun, and S.-L. Jämsä-Jounela. Control reconfiguration in networked control system. In *Proc. of 6th IFAC Symp. on Fault Detection and Safety of Technical Processes*, 2006.