Dependability of switched network architectures for Networked Control Systems

Sylvain Kubler, Eric Rondeau, Jean-Philippe Georges

Dependability of switched network architectures for Networked Control Systems

Sylvain Kubler and Éric Rondeau and Jean-Philippe Georges
Research Centre for Automatic Control of Nancy,
Nancy-University, CNRS, UMR 7039
Campus Science BP 70239, F-54506 Vandœuvre lès Nancy, France.
FirstName.Name@cran.uhp-nancy.fr

***Abstract*—The switched Ethernet networks are more and more deployed in industry. The Spanning Tree Protocol implemented in the switches enables to manage the link connectivity. But the reconfiguration time of Spanning Tree Protocol (STP) when link failure occurred is not adapted in regard to the industrial constraints. The objective of this paper is to propose a method based only on standard mitigating the probability of disconnection between nodes having hard real-time properties. The approach developed in this paper consists in duplicating frames and in forwarding them on different paths. These paths are optimized and specified by using genetic algorithms. OPNET simulations show the interest of this proposal on a particular Networked Control System.**

I. Introduction

The modern plant architectures implement communication networks to control and to monitor their remote and distributed applications. The major interest facing the point to point architectures is to mitigate the wire costs and to make easier the information sharing. In the context of embedded systems and industrial systems, the communication requirements are both to guarantee bounded end-to-end delays and to maintain the connectivity between all the network nodes. In the 1980's, many networks named fieldbuses were developed to respect these strong constraints. More recently, the trend in fieldbuses consisted in using Ethernet protocol (IEC 61784 gathers the different fieldbuses standards). The advantages are Ethernet is a well known protocol, widely implemented (ensuring its permanence), and its performance are continually increasing with the technologies evolutions (especially its bandwidth). But, Ethernet is not a determinist network since it uses the CSMA/CD protocol to access to the medium. However, the full switched architectures (IEEE 802.1d) allow inhibiting the collisions and are able to manage redundant links. The Spanning Tree Protocol (STP) enables to reconfigure the physical network architecture when a link is in failure in using redundancies. Nevertheless, the standard STP reconfiguration time period is too long and is not compatible with the industry requirements. The objective of this paper is to propose a method based on STP standards for reducing the probability of disconnection between nodes with critical application constraints. The practical approach defended in this paper, is to duplicate the messages with real-time properties and to send them to the remote nodes by using several paths. If one path fails, messages will still arrive at destination by another path in hiding the STP reconfiguration time period on the trouble path.

This article is organized in the following way: the section II describes the Spanning Tree Protocol standards and the related works. The section III formalizes the objective function to be optimized. The section IV presents the approach based on genetic algorithm to increase the dependability of switched network architectures. Finally, the section V illustrates the interest of this study in the Networked Control Systems framework.

II. Spanning tree

A. Introduction

Basically, the dependability on network wire is achieved by implementing redundant links and/or network devices. In Ethernet, the redundancy may generate loops in the physical architecture which can lead to infinite retransmissions consuming needlessly bandwidth. The Spanning Tree Protocol (STP) implementing in the switches, aims at breaking the loops in order to form a tree by disabling some links. STP has also to maintain the network connectivity. Then, STP continually monitors the physical link, and dynamically adjusts their states (enable, disable) according to the failure occurrences. Moreover, MSTP (Multiple STP, IEEE 802.1s) or PVST (Per-Vlan STP, Cisco proprietary protocol) are protocols offering the possibility to define as many trees as Virtual Networks (VLAN, IEEE 802.1q).

B. Reconfiguration time

As described previously, the problem of STP is the reconfiguration time period to propose a new fit tree which is around 30 s. This time period corresponds to the addition of failure detection time, spanning tree computation time and link state change time. The new Rapid STP (RSTP, IEEE 802.1w) mitigates this reconfiguration time and enables to propose new physical architecture is less than 5 s. However, these times may still not be adapted to real-time environments. The MOXA Company has then designed a new mechanism named *Turbo-ring*, which ensures an Ethernet reconfiguration time period around 20 ms. But this solution is not standardized, and it is usable only on a particular topology (a ring), and finally it still generates a breaking time. The objective of this paper

is to define a method avoiding this discontinuity of service by even leading it to zero.

C. Related works

In [1], an on-line method which implements a new disjoint multipath routing algorithm (*SimCT*) is defined in packet-switched networks. This work is based on the colored graph trees theory and takes into account the nodes failures, in order to maintain the communication continuity between a node and a sink. The interest of [1] is to mitigate the long path compared with others proposals and to reduce the number of protocol frame exchanges. Another approach consists in anticipating off-line the path failure. [2] works on Ethernet solutions without switches (hub architecture) and proposes to duplicate the medium and to send the messages twice respectively on the two mediums. A mechanism named *link selector* is implemented inside each node to ensure the incoming information consistency and to duplicate the sent messages. [1] and [2] are related works, but our approach differs relatively to the network technology. First, we study switched Ethernet architectures not considered in [1] [2]. Secondly, our main objective is to maintain the continuity of service in respecting the standards defined for switched Ethernet. Especially, the MSTP protocol is used to define several trees for interconnecting Networked Control System (NCS) equipment. Also, the strategy consists in implementing a static procedure which consists in forwarding as many frames as defined trees. Thus, if a path is down, the sink node receives the information by at least one of the others paths.

III. Formalization

A. Introduction

The first work is to define an expression formalizing the problem of the redundant paths between a source generating traffic and its destination node. A path consists of a series of network components. A network component represents either a network node (switch) or a link. The failure probability of a path depends on the number of network components constituting the path. More the network components are used, more the path failure probability is high. Moreover, the failure probability of the redundant paths can be cumulated when a same network component is used in several paths. Indeed, while this network component fails, it may affect the behavior of not only one path but of several paths. Thus, the network dependability evaluation depends on the path length (section III-B) and on the number of common network components implemented in several paths (section III-C). In the present paper, each network component is assumed to have the same failure probability.

B. Network dependability with independent paths

We define α_i as being the number network components composing the path i. Let λ be the failure probability per hour of any network component and $\mu = 1 - \lambda$ as being the non-failure probability. The failure probability of a path i (P_{Path_i}) is given by (1). The failure probability of the network (P_{Net_j}) composed of j independent paths is given then by (2).

$$P_{Path_i} = 1 - (1-\lambda)^{\alpha_i} = 1 - \mu^{\alpha_i} \quad (1)$$

$$P_{Net_j} = \prod_{i=1}^{j} P_{Path_i} = \prod_{i=1}^{j} (1 - \mu^{\alpha_i}) \quad (2)$$

The assessment of (2) is studied according to the SIL specifications (Safety Integrity Level, IEC 61508 standard). The Table I provides the values used in the continuous mode. Our research focuses on Networked Control Systems where the SIL 3 and SIL 4 are the levels generally required, especially in avionic, nuclear power plant,... The analysis is then based on the choice of network components with SIL 4 (with $\lambda = 10^{-8}$). The aim is to observe whether the whole network (constituted of SIL 4 network components) respects the SIL 4 constraint according to both the number of paths and the number of network components used inside a path.

TABLE I
DEFINITION OF SIL IN THE CONTINUOUS MODE (IEC 61508)

SIL	Range of λ (Failures by hour)
4	$10^{-9} < \lambda < 10^{-8}$
3	$10^{-8} < \lambda < 10^{-7}$
2	$10^{-7} < \lambda < 10^{-6}$
1	$10^{-6} < \lambda < 10^{-5}$

The results are given in Fig. 1. The first observation is the network with one path is quickly degraded and the scores are upper SIL 4. The second comment is that a network with two independent paths gives a good result since the SIL 4 is guaranteed up to 10 000 network components. Finally, the multiplication of paths considerably increases the cost of the network in terms of price, maintenance, complexity,... without necessarily improving the solution relatively to SIL constraints. In conclusion, a network architecture implementing two paths is a good arrangement between dependability and cost. Thereby, the study only analysis redundancy with two paths in the next section.

C. Paths with overlapping

In this section, a network consists of two paths with overlapping. Let α_i be the number of exclusive network components composing the path i. An exclusive network component belongs only to one path. Let β be the number of common network components used in both paths. Thus, the total length of a path i is equivalent to $\alpha_i + \beta$. By relying on the equation (2), the whole network failure probability (P_{Net}) can be determined by (3).

$$P_{Net} = 1 - \big(1 - (1-\mu^{\alpha_1})(1-\mu^{\alpha_2})\big)\mu^{\beta} \quad (3)$$

Fig. 2 gives the failure probabilities defined in (3) according to the path length and the rate of common network components used in the two paths : $\beta/(\alpha_i + \beta)$. Fig. 2 shows that when at

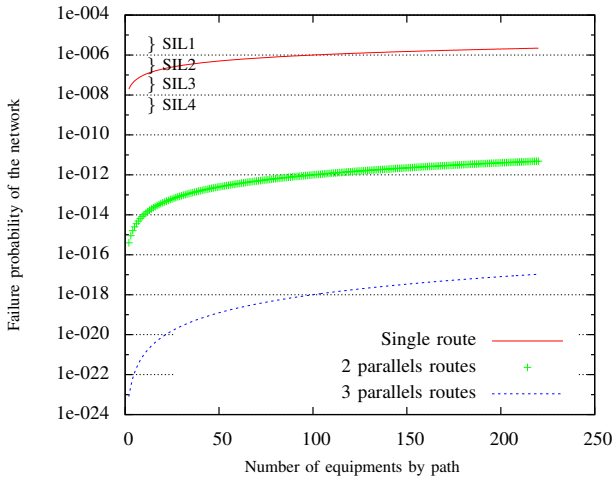


Fig. 1. Analysis of dependability for independent paths

least one of common network components fails, the whole network is strongly affected. As a result, performance are immediately under the SIL 4 constraint.

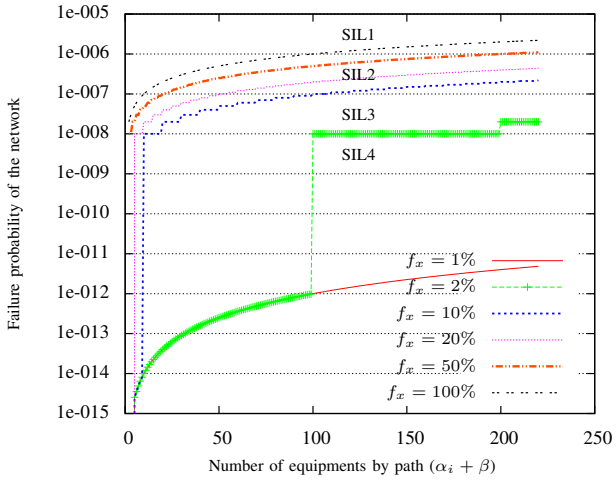


Fig. 2. Objective function for two paths, according to an overlapping percentage $f_x = \frac{\beta}{\alpha_i + \beta}$

D. Conclusion

The expression (3) is a general expression of (2) with two paths. (3) is then used as objective function to find optimized pair of trees. The network consists of many links, many nodes such that the search of pair of trees quickly meets an explosion combinatory problem. This issue is a subclass of problem studied in [1] described with complexity order $O(|L||N|)$, with $|L|$ the number of links and $|N|$ the number of nodes. Thereby, we propose as suggested in [1] to use heuristic algorithms.

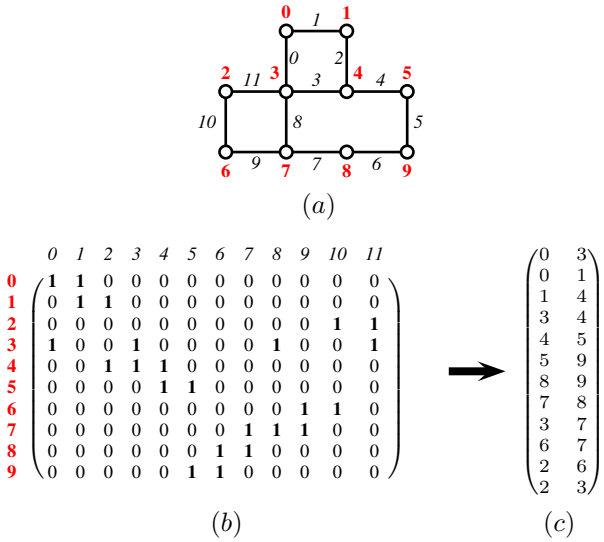
IV. Optimization algorithms

A. Choice of heuristic

There are three main classes of heuristics: the constructive methods (greedy algorithm, pilot method), the local research methods (simulated annealing, taboo search) and the evolutionary methods (Genetic Algorithm, ant colony optimization). It is difficult to compare these methods. But, an analysis in [3] between the simulated annealing, the taboo search, the ant colony optimization and the genetic algorithms (GA) provides diagrams giving the probability that a method is better than another one. In these works, we notice for a similar problem, the GA give the best results to this type of study.

B. Coding of the component networks

The network architecture is defined by a graph $\mathcal{G} = (\mathcal{S}, \mathcal{M})$ (digraph or undirected graph) where $\mathcal{S}$ is the set of nodes, i.e. the vertices of the graph and where $\mathcal{M}$ is the incident matrix and defines the network links. As a matter of fact, the matrix $\mathcal{M}$ stands for the "incident-vertex-edge matrix" of dimension $m \times n$ where m is the number of vertices and n is the number of edges. Each edge of the graph is numbered such that considering an edge i interconnecting two vertices k and l, we have $\mathcal{M}(j,i) = 1 \forall j = k, l$ and $\mathcal{M}(j,i) = 0$ otherwise. As shown in Fig. 3, this matrix can be summarized in a vector where each row corresponds to one edge of $\mathcal{G}$ and where columns give the edges of a given vertex.



(a) (b) (c)

Fig. 3. Network modeling, (a) graph $\mathcal{G} = (\mathcal{S}, \mathcal{M})$, (b) incident matrix $\mathcal{M}$ and (c) its equivalent vector

According to this definition, it is possible to propose two algorithms aiming at checking if the graph is connected and identifying the nodes along a path. There are two types of algorithms for going through a graph, the Depth First Search (DFS) and the Breadth First Search (BFS). The difference is related to the analysis sequence of the vertices. The DFS explores "deeply" the paths one by one, whereas the BFS generates a research tree achieving layer by layer. However, the BFS requires a lot of memory to store all the alternatives for each layer. Therefore, the DFS algorithm is implemented in our work.

Firstly, an algorithm is defined in order to check that a vertex s of the graph $\mathcal{G}$ is able to access any vertices of $\mathcal{S}$. Running `Accessible_Nodes` $(\mathcal{G}, s, \{\emptyset\})$ from Algorithm 1 returns the list $\mathcal{V}$ of the accessible vertices from s. Hence if $\mathcal{V} = \mathcal{S}$, it means that a graph is connected. This algorithm will then use to check that the proposed trees are respectful according to the spanning tree definition.

Algorithm 1: `Accessible_Nodes` $(\mathcal{G}, i, \mathcal{V})$

output: the list $\mathcal{V}$ of the accessible nodes from i

begin
- $\mathcal{V} \longleftarrow \mathcal{V} \cup \{i\}$;
- **forall the** $j \in \mathcal{S} \backslash \mathcal{V}$ **do**
  - **if** $\exists k \mid \mathcal{M}_{i,k} = \mathcal{M}_{j,k} = 1$ **then**
    - `Accessible_Nodes` $(\mathcal{G}, j, \mathcal{V})$;

Secondly, the function `Path` $(\mathcal{G}, i, e, \mathcal{V}, \mathcal{P})$ of Algorithm 2 is proposed in order to compute the path length. Indeed running `Path` $(\mathcal{G}', s, e, \{\emptyset\}, \{\emptyset\})$ returns a list $\mathcal{P}$ containing the vertices of the graph $\mathcal{G}' = (\mathcal{S}, \mathcal{M}')$ along the path between s and $e \in \mathcal{S}$. Hence, the path length is given by the size of $\mathcal{P}$. This function will be used in the following to compute the objective function of a spanning tree solution $\mathcal{G}'$.

Algorithm 2: `Path` $(\mathcal{G}, i, e, \mathcal{V}, \mathcal{P})$

output: the list $\mathcal{P}$ of the nodes along the path from e to i

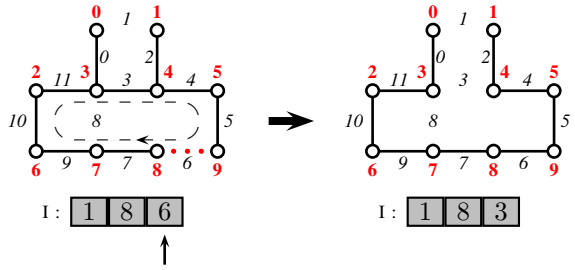
begin
- $\mathcal{V} \longleftarrow \mathcal{V} \cup \{i\}$;
- **if** $i = e$ **then**
  - $\mathcal{P} \longleftarrow \mathcal{P} \cup \{e\}$;
- **else**
  - **forall the** $j \in \mathcal{S} \backslash \mathcal{V}$ **do**
    - **if** $\exists k \mid \mathcal{M}_{i,k} = \mathcal{M}_{j,k} = 1$ **then**
      - `Path` $(\mathcal{G}, j, e, \mathcal{V}, \mathcal{P})$;
  - **if** $\mathcal{P} \neq \{\emptyset\}$ **then**
    - $\mathcal{P} \longleftarrow \mathcal{P} \cup \{i\}$;
- **return** $\mathcal{P}$;

Finally, the last step deals with the coding of one spanning tree. There are several methods of spanning tree representation of graph [4], [5], [6], [7], [8], [9], [10]. [11] explains by taking the method of coding based on the representation in co-tree (Digital coding of open branches), it avoids the determination of the fundamental loops allowing to test the validity of the resultant topologies after crossover and mutation processes in the genetic algorithms. In conclusion, the "digital coding of open branches" method [7] is implemented. For instance, considering the graph Fig. 3, a possible spanning tree could be modeled by the sequence $1, 8, 6$ in which each number represents the opened branch as shown in Fig. 4

C. Genetic Algorithm

According to observations based on the evolution of species, C. Darwin introduces the theory of evolution.

In this paper, individual represents the spanning tree. According to the type of coding previously cited, the crossover and mutation operators might generate invalid individuals compared to the topological constraints. [11] developed a crossover and mutation strategy based on the graphs theory approaches. The mutation technique consists in randomly picking a gene from a chromosome (representative of an opened branch), and then to determine the formed loop resulting of the closure of this branch. Finally GA randomly selects within this loop a branch to open. This principle is illustrated Fig. 4.



Step 1 : Random choice of one gene (branch) to mute

Step 2 : Random choice of one branch resulting from the formed loop next the closure of the branch 6. Loop : $[3, 4, 5, 6, 7, 9, 10, 11]$

Fig. 4. Mutation technique

The crossover technique illustrated on Fig. 5, consists in randomly picking a reference point in the chromosome of the individual I1, and then to determine the loop formed after the closure of this branch. GA looks for branches which might be swapped between the 2 co-trees. Moreover, it is necessary to analyze the chromosome of the individual I2. If there are genes belonging to the loop, GA has to make the crossing with one of them. It is necessary to repeat this step until reaching the last gene.

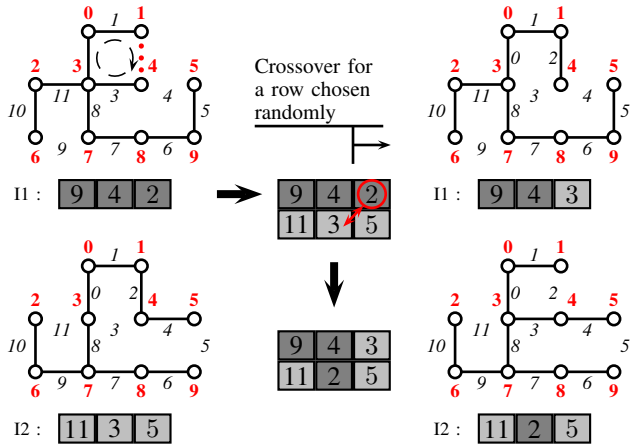


Fig. 5. Crossover technique

D. Adjustments of GA parameters

[12] defined conventional GA parameters such as population size t_{pop}, crossover p_c and mutation p_m probabilities. These parameters have an influence on the GA performance. It is then necessary to study the impact of each probability on GA behavior to determine their optimal value. [13] proposes an approach allowing to achieve these adjustments. This method is used in this research and the optimal values obtained are: $t_{pop} = 10$, $p_c = 70\%$ and $p_m = 40\%$ [14].

V. Case of study

A. Introduction

Firstly, specific nodes were developed in the OPNET network simulation tool, to implement the behavior of NCS device. The NCS implemented in our simulations is depicted in Fig. 6. To give numerical results, the approach presented in this paper will be applied on a sample system described by its following system equations:

$$P(s) = \frac{2}{(s+5)(s+0.2)}, C(s) = \frac{0.5508s + 0.4529}{s}$$

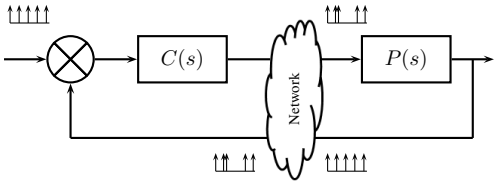


Fig. 6. NCS feedback control loop

Secondly, a particular switched Ethernet architecture consists of 31 switches and 56 links is defined Fig. 7. The NCS devices (controller, sensor, actuator) are respectively connected to switches 20, 5 and 11. The cycle time is 2 ms, the devices periodically send packets of 64 bytes every 1 ms.

Thirdly, the study is divided in two steps. The first one uses the classic solution with the Spanning Tree Protocol (STP). According to the values of switches and ports priorities, the STP algorithm will defined one spanning tree for the whole network. Fig. 7 depicts the default spanning tree observed in our network. The second one applies the method proposed in this paper, in implementing two paths obtained from GA. Rapid STP is activated on the simulation tool for defining the two trees. To improve the understanding of the figures, only the links belonging to the paths between the NCS devices are highlighted (bold lines) and not the global tree.

B. NCS analysis with RSTP

Fig. 7 shows the path used by the controller to communicate both with its sensor and actuator. This path is a part of one tree that can be defined by RSTP for a particular set of switches and ports identifiers. During the simulation, the link 8 belonging to this path fails at $t = 115$ s. Fig. 9(a) provides the behavior of the process output according to the reference. Before the link failure, the process reaches the reference with an acceptable overshot. When the link 8 fails, the actuator is disconnected to the controller during 5 s corresponding to the RSTP reconfiguration time period. Thus, between $t = 115$ s and $t = 120$ s, the new references are not taken into account by the process. After $t = 120$ s, the controller is able to send to the actuator the new control, but a period of instability is observed before reaching again the reference. This instability step may be due to the reception of old messages (by the actuator) buffered in the network.

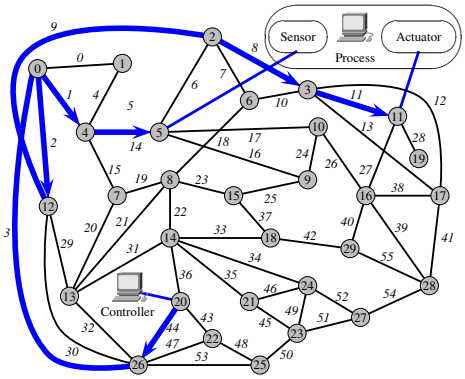


Fig. 7. Classic STP solution

C. NCS analysis with two redundant paths determined by GA

Fig. 8 shows the two independent paths obtained by GA for interconnecting the controller and the process. Two VLANs are defined and two trees are created respectively for each VLAN by using MSTP. In technical way, the network designer has only to set the switch ports belonging to the path with a high priority, and the root switch is located at the middle switch of the path. Consequently, MSTP determines its tree in overlapping the path. In this approach, the controller duplicates the control messages on the two trees and it receives twice the sensor state.

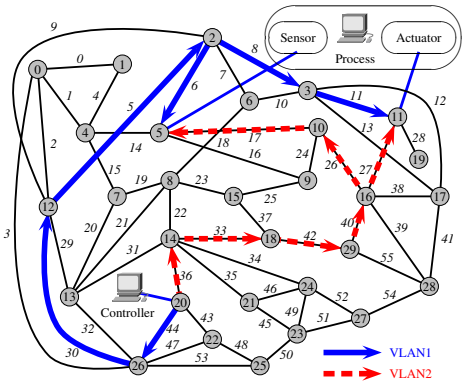
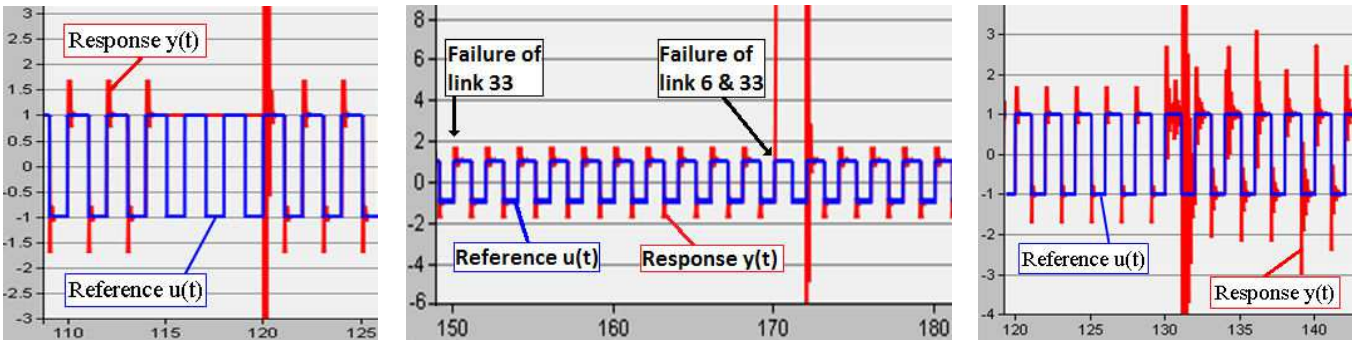


Fig. 8. Duplication of messages on a Multi-VLAN architecture

When the link 33 fails at $t = 150$ s, the Fig.9(b) shows the process carries on to work since only the path associated to VLAN 2 is broken and the connectivity is maintained by the



(a) System output for the classic STP solution (b) System output in case of duplicated messages (c) System output without packets numbering

Fig. 9. Simulation of the Network Control System behavior

second path associated to the VLAN 1. Next, when both links 6 and 33 fail at $t = 170\ s$, the system becomes unstable. However, this occurrence appears with a probability at SIL 4 if all the network components are SIL 4 as it is explained in the section III.

D. Temporal validity of messages

The classical problem due to the message duplication is the consistency control of information system. The duplication generates desynchronizations since the messages duplicated do not arrived at the same time on the destination nodes. Fig. 9(c) illustrates this problem by adding load traffic at $t = 130\ s$ on the link 42 (path VLAN 2). This overload generates a congestion and some delays on messages which are sent on the path VLAN 2. Thus, the messages sent on path VLAN 1 arrive before the ones using the path VLAN 2. Fig. 9(c) shows instabilities since the controller processes the received messages in FIFO order, without managing their temporal validity. This trouble can be suppressed either by numbering or timing the messages. The numbering of messages is implemented in our OPNET simulations and thereby, the reception nodes discard all the messages containing "old" number. This method is applied to eliminate the instabilities issues. The results are similar to the Fig.9(b).

VI. Conclusion

The paper presents both a strategy to face with high dependability constraints and an off-line method to define in practise the pair of paths for each traffic based on a local configuration (without modifying Ethernet standards). This method is based on the definition of two redundant paths determined by GA and relies on message duplication thankfully the MSTP technique. The main interest is to reduce the probability of communication disconnection in particular in a NCS framework. However, this method has to be applied only for specific nodes strongly timed constrained. It cannot be generalized for all the nodes of the network, since the duplication of messages induced overload.

Acknowledgement

This work was supported by OPNET Technologies, Inc. thanks to "Teaching with OPNET" program.

References

[1] G. Jayavelu, S. Ramasubramanian, and O. Younis, "Maintaining colored trees for disjoint multipath routing under node failures," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 346–359, 2009.

[2] S. Limal, S. Potier, B. Denis, and J.-J. Lesage, "Formal verification of redundant media extension of Ethernet PowerLink," in *Proceedings of 12th IEEE Conference on Emerging Technologies and Factory Automation 12th IEEE Conference on Emerging Technologies and Factory Automation*, Patras Grèce, 09 2007, pp. pp. 1045–1052.

[3] J. Dréo, *Metaheuristics for hard optimization : methods and case studies*. Springer: Springer, 2006.

[4] B. Enacheanu, B. Raison, D. Ivanova, R. Caire, A. Aubry, and N. Hadjsaid, "Flexible Electric Infrastructures for Advanced Distribution Automation," in *CRIS 2006, Third International Conference on Critical Infrastructures*, Alexandria États-Unis d'Amérique, 09 2006.

[5] K. Nara, A. Shiose, M. Kitagawa, and T. Ishihara, "Implementation of genetic algorithm for distribution systems loss minimum reconfiguration," *IEEE Transactions on Power Systems*, vol. 7, no. 3, pp. 1044–1051, Aug 1992.

[6] Y. Zhu and K. Tomsovic, "Adaptive power flow method for distribution systems with dispersed generation," *Power Engineering Review, IEEE*, vol. 22, no. 5, pp. 72–72, May 2002.

[7] B. Radha, R. T. F. A. King, and H. C. S. Rughooputh, "A modified genetic algorithm for optimal electrical distribution network reconfiguration," in *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, R. Sarker, R. Reynolds, H. Abbass, K. C. Tan, B. McKay, D. Essam, and T. Gedeon, Eds., vol. 2. Canberra: IEEE Press, 8-12 December 2003, pp. 1472–1479.

[8] E. R. Ramos, A. G. Exposito, J. R. Santos, and F. L. Iborra, "Path-based distribution network modeling: application to reconfiguration for loss reduction," *Power Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 556–564, 2005.

[9] W.-M. Lin, F.-S. Cheng, and M.-T. Tsay, "Distribution feeder reconfiguration with refined genetic algorithm," *IEEE Transactions on Power Systems Proceedings - Generation, Transmission and Distribution*, vol. 147, no. 6, pp. 349–354, 2000.

[10] Y.-Y. Hong and S.-Y. Ho, "Determination of network configuration considering multiobjective in distribution systems using genetic algorithms," *Power Systems, IEEE Transactions on*, vol. 20, no. 2, pp. 1062 – 1069, May 2005.

[11] B. Enacheanu, B. Raison, R. Caire, O. Devaux, W. Bienia, and N. Hadjsaid, "Radial network reconfiguration using genetic algorithm based on the matroid theory," *Power Systems, IEEE Transactions on*, vol. 23, no. 1, pp. 186–195, 2008.

[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[13] 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*, vol. 19, no. 3, pp. 335–344, 04 2006.

[14] S. Kubler, E. Rondeau, and J.-P. Georges, "Continuité de service sur Ethernet Industriel," in *Sixième Conférence Internationale Francophone d'Automatique, CIFA 2010*, 06 2010.