



Evaluation of Response Time in Ethernet-based Automation Systems

Gaëlle Marsal, Bruno Denis, Jean-Marc Faure, Georg Frey

► To cite this version:

Gaëlle Marsal, Bruno Denis, Jean-Marc Faure, Georg Frey. Evaluation of Response Time in Ethernet-based Automation Systems. 6th IEEE International Workshop on Factory Communication Systems (WFCS'2006), Jun 2006, Turin, Italy. pp.95-98. <hal-03746352>

HAL Id: hal-03746352

<https://hal.science/hal-03746352v1>

Submitted on 5 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Evaluation of Response Time in Ethernet-based Automation Systems

Gaëlle MARSAL, Bruno DENIS,
Jean-Marc FAURE
LURPA - ENS de Cachan
61, av. du Président Wilson
94235 Cachan cedex - France
gaelle.marsal@lurpa.ens-cachan.fr

Georg FREY
Department of Electrical and Computer
Engineering - University of Kaiserslautern
Erwin-Schrödinger-Str. 12
67653 Kaiserslautern - Germany
georg.frey@eit.uni-kl.de

Abstract

This paper presents ongoing works on evaluation of response time of switched Ethernet-based automation systems. Two methods are proposed to evaluate the response time: summation of elementary delays and simulation of hierarchical colored timed Petri net models. Comparison of the results yielded by these two methods on an application example permits to show the advantages of the simulation approach.

1. Introduction

Automation component vendors today propose Ethernet-based solutions to connect logic controllers to remote input-output modules (RIOMs). These new automation architectures (Figure 1) generally include switches to avoid collisions and, provided that the size of internal buffers of these switches are large enough, frame loss. This hypothesis about the absence of frame loss will be assumed true in the rest of this paper. Logic controllers include at least two modules: one CPU module that executes the user program and one Ethernet module that scans RIOMs. These modules communicate data via a backplane bus and a shared memory area.

To control communication between the different components of these architectures (Ethernet modules of the PLCs, switches and RIOMs), specific protocols have been developed, like Profinet [1] or DP-Ethernet [2], that rely upon master-slave or producer-consumer models. Other vendors, like Schneider Electric with its Modbus/TCP solution, prefer to use the standard Ethernet client-server protocol. Only this last kind of protocol will be addressed in this paper.

The main advantage of these client-server switched Ethernet architectures is to allow devices that use the standard Ethernet protocol to communicate directly, for monitoring and diagnosis purposes for instance, with RIOMs and consequently with the process. Despite this interesting feature, design of these architectures requires to consider their response time in detail. As communication control is no more centralized, the response time,

delay between the occurrence of an input event and the occurrence of the corresponding output event (Figure 1) is not easy to evaluate from the definition of the architecture. RIOMs and switches are shared resources indeed and the delays they introduce depend on their load. Moreover, periodic user program execution, RIOMs scanning by the Ethernet module as well as input events generation are asynchronous processes; synchronization between these different processes induces supplementary delays. Hence response time evaluation in the studied architectures deserves careful analysis that must take into account different kinds of delays.

This paper presents results of ongoing works that are aiming at proposing a method to assess the response time of client-server switched Ethernet-based automation architectures by simulation of hierarchical colored timed Petri nets (HCTPNs). The following section details the causes of delays in the different components of the architecture. In section 3, a simple method to obtain approximate bounds of the response time is proposed, while section 4 focuses on the Petri nets model of the architecture. An application example is presented in section 5 to illustrate the benefits of simulation for response time evaluation. Prospects on future works are given in the concluding section.

2. Response time in Ethernet-based automation systems

Figure 2 illustrates the different processes performed by the architecture components as well as the propagation of information (black arrows) from an occurrence of an input event to the occurrence of the corresponding output event. Three kinds of delays appear in this figure:

- Processing times
- Waiting times for resource availability
- Waiting times for synchronization.

Focusing for instance on the RIOM that receives the input event, this figure shows that, sequentially:

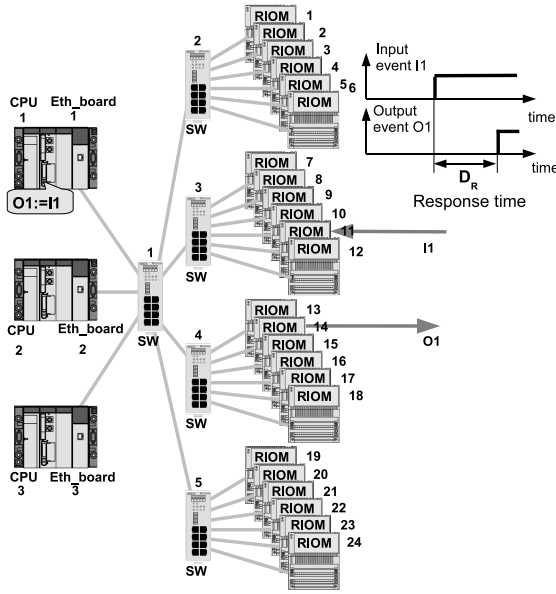


Figure 1. Response time in switched Ethernet-based automation system

- this event is first filtered to avoid unexpected high frequency noise (processing time),
- then the RIOM waits for a request from an Ethernet board (waiting time for synchronization),
- this request has to wait into an internal queue of the availability of the RIOM internal processor (waiting time for resource availability),
- and finally the RIOM constructs a frame that contains the event and sends this frame (processing time again).

The processing time of this component is the sum of the times for filtering, frame construction, and sending. Therefore, the total delay induced by this RIOM is the sum of the three kinds of delays discussed above. It matters to highlight that all components do not behave in a similar fashion. Switches induce only processing time and waiting time for resource availability. This latter one is not present for CPU modules of controllers, if only mono-tasking scheduler is considered. Moreover the delays decomposition for Ethernet and CPU modules depends on the direction of information flow. It is not the same when information flows from the process to the controller and in the other direction.

Table 1 sums up the elementary delays caused by each component. The response time D_R is obviously the sum of all the delays caused by the components that are crossed. So:

$$D_R = d_{RIOM}^1 + d_{SW}^1 + d_{EthM}^1 + d_{CPU}^1 + d_{EthM}^2 + d_{SW}^2 + d_{RIOM}^2$$

Where:

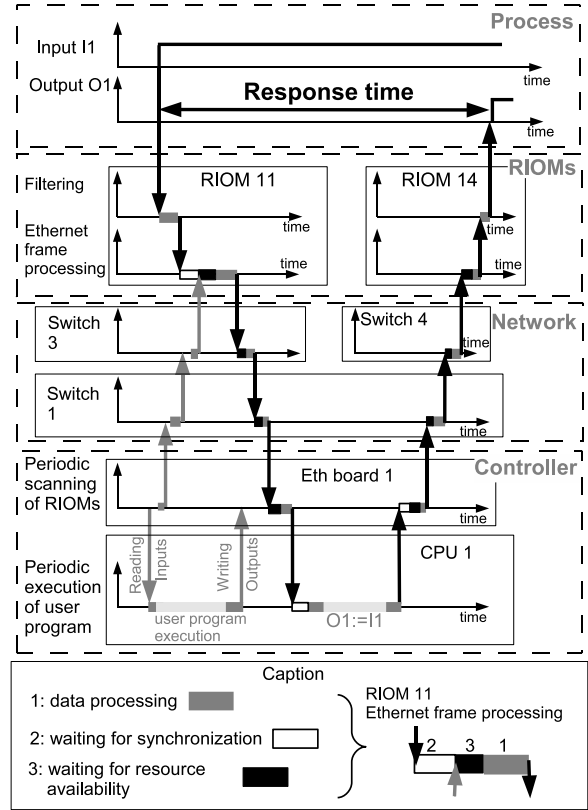


Figure 2. Response time decomposition

Table 1. Elementary delays caused by

	Processing	Waiting for resource availability	Waiting for synchronization
RIOM	d_{pRIOM}	d_{wRIOM}	d_{sRIOM}
SW	d_{pSW}	d_{wSW}	0
EthM	d_{pEthM}	d_{wEthM}	d_{sEthM}
CPU	d_{pCPU}	0	d_{sCPU}

- d_{XYZ}^1 represents the delay caused by the component XYZ when information flows from the process to the controller
- d_{XYZ}^2 represents the delay caused by the component XYZ when information flows from the controller to the process.

3. Response time evaluation by summation of elementary delays

Processing times are constant but waiting times for resource and for synchronization are variable and not independent. Increasing the waiting time for resource within a component for instance can lead to decrease or increase the waiting time for synchronization of this component or of an other component and conversely. Then, analytic computation of the response time of the whole architecture is a tough objective. Nevertheless, a rough evaluation of the bounds of the response time can be obtained, provided that the two following assumptions would be made:

- waiting times for resource availability are neglected
- delay causes are independent.

Given these assumptions, the response time is a sum of processing times, that can be obtained from technical documentations or from experiments, and of waiting time for synchronization. The lower bound is then obtained when all the processes are synchronous (all synchronization times equal zero) and is therefore:

$$D_{Rlb} = d_{pRIOM}^1 + d_{pSW}^1 + d_{pEthM}^1 + d_{pCPU}^1 + d_{pEthM}^2 + d_{pSW}^2 + d_{pRIOM}^2$$

The upper bound is obtained when synchronization times are maximal, that gives :

$$D_{Rub} = d_{pRIOM}^1 + d_{sRIOM}^1 + d_{pSW}^1 + d_{pEthM}^1 + d_{pCPU}^1 + d_{sCPU}^1 + d_{pEthM}^2 + d_{sEthM}^2 + d_{pSW}^2 + d_{pRIOM}^2$$

4. Response time evaluation by simulation of hierarchical colored timed Petri nets

Simulation of Petri nets for the analysis of Ethernet based systems has been addressed by many researchers (for instance [3]) who used different classes of Petri nets. As mentioned in the introduction, it is assumed that no frame loss occurs in the considered architectures. Components failures are also out of the scope of this paper. This explains why stochastic Petri nets [4], a widely-used formalism for performance evaluation of networked systems have not been selected. The same holds for other probabilistic approaches like probabilistic timed automata [5].

The objective being evaluation of response time, time or timed Petri nets may be considered. The latter ones have been chosen for internal processing times are known with a quite good precision; hence it is not necessary to model uncertainty about duration with time intervals. The architecture model will include classical structures to model resource allocation and synchronization; waiting times will be obtained from simulation of these structures. At last, to ensure genericity of our model, colored timed Petri nets [6] have been selected.

This approach is similar to that presented in [7]. Compared to these works, the generic model that we have developed [8] includes places and transitions that model resource availability; this enables a more detailed analysis taking into account components load.

For space reasons, only the highest hierarchical level of this model is sketched here (Figure 3) where each type of component appears as a transition which substitutes a Petri net model. For instance, beside the transition “Ethernet modules of PLC” is modeled the Ethernet module behaviour, and in particular the periodic scanning of RIOMs. It matters to notice that the number of components,

their configuration and their connection are parametrisable. The interface places contain tokens to model information flows between the components. These tokens can represent either events or Ethernet frames.

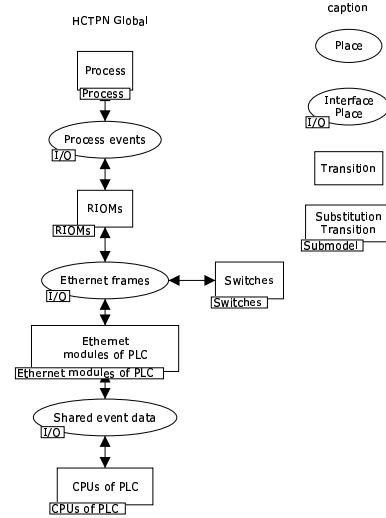


Figure 3. Highest hierarchical level of TCPN Architecture model

5. Application example

The two previously presented approaches (addition of elementary delays and simulation of HCTPNs) have been applied to the architecture of Figure 1 with the values given in Table 2. The values of controllers CPUs periods of user program execution and of Ethernet modules scan periods are given with one millisecond resolution so as to avoid unexpected synchronizations when simulating the Petri net model.

In the first configuration, scan periods are long and all RIOMs, that act as data servers, are shared by at least two clients, RIOMs 11 to 14 being even shared by three clients. This configuration will therefore lead to considerable waiting times both for synchronization and for resource availability. In the second configuration, scan periods are shorter and resource sharing is less critical; only four RIOMs (8 and 9, as well as 16 and 17) are shared by two clients. This configuration is more reactive and shall lead a priori to smaller waiting times.

The results obtained with the two methods are presented in Table 3. The results related to the second method are the minimum and maximum response times obtained during 8000 simulations of the Petri net model. Although simulation permits to provide a distribution of the response time, comparison of the two evaluation methods is indeed only possible on the basis of minimum and maximum values.

The results obtained for the first configuration show that waiting times for resources do not influence strongly the

Table 2. Time configuration in milliseconds

	conf. 1	conf. 2
CPU 1 period	18.000	7.000
CPU 2 period	16.003	7.001
CPU 3 period	20.005	9.000
Scan period of Eth module 1	45.000	5.500
Scan period of Eth module 2	45.003	17.001
Scan period of Eth module 3	120.007	13.003
Switch processing delay	0.1	
RIOM processing delay	0.72	
Eth board processing delay	0.15	
RIOM scanned by Eth module 1	1 to 14	1 to 9
RIOM scanned by Eth module 2	11 to 24	8 to 17
RIOM scanned by Eth module 3	1 to 24	16 to 24
Program (OI:=I1) executed by	CPU 1	
I1 read from	RIOM 11	RIOM 8
O1 updated to	RIOM 14	RIOM 9

Table 3. Minimum and maximum response time evaluated by summation and HCTPN models (milliseconds)

	conf. 1		conf. 2	
	min	max	min	max
Sum model (1)	20.14	128.14	9.14	27.14
HCTPN model (2)	44.77	92.06	11.85	34.97
deviation (%) of (1)/(2)	55.01	39.19	22.87	22.39

overall response time when waiting times for synchronization are significant. In this case, even if many resources are shared, the first assumption introduced in section 3 (waiting times for resources can be neglected) holds true. The first evaluation method provides real lower and upper bounds.

On the contrary, the results obtained for the second configuration pinpoints clearly the weakness of this method: the maximal value of the response time that it provides is not an upper bound when waiting times for synchronization are small.

The first evaluation method can therefore only be used for architectures in which waiting times for synchronization are long. If this condition is not satisfied, only simulation of the Petri net model that does not rely on any assumptions yields realistic results.

6. Conclusion

Our current results have demonstrated the interest of simulation of hierarchical colored timed Petri nets for evaluation of response time of client-server switched Ethernet automation architectures. To validate the developed model, experiments on real architectures are planned. Comparison of the experimental results to the simulated ones will permit either to refine the existing model or to increase the level of confidence in simulation results.

Once the model is validated, it will be possible to identify the parameters that impact most the response time and consequently to propose rules to design efficient architectures that comply with specified response times.

References

- [1] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, "Experimental evaluation of PROFINET performances", in *5th IEEE Int. Workshop on Factory Communication Systems, WFCS04*, September 2004, Vienna, Austria.
- [2] S. Vitturi, "DP-Ethernet: the Profi bus DP protocol implemented on Ethernet", *Computer Communications*, vol. 26, pp. 1095–1104, 2003.
- [3] J. de Figueiredo and L. Kristensen, "Using Coloured Petri Nets to Investigate Behavioural and Performance Issues of TCP Protocols", in *CPN Workshop, CPN'99*, October 1999.
- [4] N. Dingle, P. Harrison, and W. Knottenbelt, "Response Time Densities in Generalised Stochastic Petri Net Models", in *3rd Int. Workshop on Software and Performance, WOSP 2002*, 2002.
- [5] J. Greifeneder and G. Frey, "Probabilistic Delay Time Analysis in Networked Automation Systems", in *Proc. of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2005*, September 2005, pp. 1065–1068.
- [6] K. Jensen, *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, volume 1, Basic Concepts, Springer-Verlag, monographs in theoretical computer science edition, 2nd corrected printing 1997.
- [7] D. A. Zaitsev, "Switched LAN simulation by colored Petri nets", *Mathematics and Computers in Simulation*, vol. 65, no. 3, pp. 245–249, April 2004.
- [8] G. Poulard-Marsal, D. Witsch, B. Denis, J.-M. Faure, and G. Frey, "Evaluation of Real-Time Capabilities of Ethernet-based Automation Systems using Formal Verification and Simulation", in *1ère Rencontres des Jeunes Chercheurs en Informatique Temps Réel 2005, RJCITR'05*, September 2005, pp. 27–30.