



HAL
open science

Control-based algorithm for the management of IEEE 802.11e priorities within a Wireless Networked Discrete Control System

Gilbert Habib, Jean-François Pétin, Thierry Divoux

► **To cite this version:**

Gilbert Habib, Jean-François Pétin, Thierry Divoux. Control-based algorithm for the management of IEEE 802.11e priorities within a Wireless Networked Discrete Control System. Conference on Control and Fault-Tolerant Systems, SysTol'10, Oct 2010, Nice, France. pp.CDROM. hal-00527087

HAL Id: hal-00527087

<https://hal.science/hal-00527087>

Submitted on 19 Oct 2010

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.

Control-based algorithm for the management of IEEE 802.11e priorities within a Wireless Networked Discrete Control System

G. Habib, JF. Pétin, T. Divoux

Abstract—This paper deals with the evaluation of Discrete Control Systems whose implementation is distributed among Programmable Logic Controllers (PLC) and remote Input/Output devices communicating through a Wireless Network. Using wireless communication in these applications, called Wireless Networked Discrete Control System (WNDCS), offers many advantages such as a greater mobility and reduced wiring costs, but also several disadvantages such as transmission delay, jitter phenomena or loss of messages due to non deterministic features of wireless communication. These features may have a negative impact on the behaviour of the control applications, and especially on the temporal performance and dependability requirements that a control system usually has to face. This paper proposes an algorithm for the management of the IEEE 802.11e priorities with regards to control needs in order to increase the performance of the WNDCS. Efficiency of the proposed algorithm is shown using OPNET simulations of a case study.

I. INTRODUCTION

USING wireless communication within industrial control applications offers many advantages such as reducing the wiring cost, as well as facilitating the commissioning, the reconfigurability, and the mobility of the control devices. Moreover, plant operation by human operator is more efficient thanks to wireless mobile hand devices performing data analysis. However, wireless communication suffers from many drawbacks such as information loss, stochastic delay, jitter phenomena or loss of messages. These problems are mainly induced by the non deterministic features of wireless communication that may have a negative impact on the stability, the real time performance and the dependability of the control system.

This paper focuses on the study of Discrete Control Systems (DCS) whose implementation is distributed on Programmable Logic Controllers (PLC) and/or remote Inputs/Outputs devices communicating through a IEEE 802.11 wireless network. It aims at reducing the communication negative impact on the performance of a Wireless Networked Discrete Control System (WNDCS) by promoting a co-design approach that tries to manage the network quality of service with regards to the control

requirements. To attend this target:

- IEEE 802.11e [1] has been chosen for its ability to work with priorities of the communication traffics that enable to increase the network quality of service for the higher priorities,
- An algorithm that dynamically allocates the priorities on the communication traffics with regards to the control requirements and current states.

This paper is organized as follows. Section 2 introduces main features of the WNDCS and highlights the main issues that are induced by using wireless communication between control and plant devices. Section 3 presents the IEEE 802.11e standard with traffic priorities as an efficient alternative for improving the global performance of a WNDCS. Section 4 proposes a control-based algorithm to manage the IEEE 802.11e priorities by dynamically gathering communication traffics according to their criticality for the control current states. At last, section 5 illustrates the proposed algorithm using a case study and demonstrates its efficiency through OPNET simulations. Finally, conclusions and open issues for future research are discussed in section 6.

II. WIRELESS NETWORKED DISCRETE CONTROL SYSTEM: DESCRIPTION AND ISSUES

A. Description of a WNDCS

A networked control system is a distributed system where the control loops are closed through a real-time network. This paper addresses WNDCS whose control system is characterized by a discrete state space and evolutions at discrete points in time. These points correspond to occurrences of asynchronous generated discrete events (event driven). A WNDCS is a system composed of three main components: Controller, Wireless Network and Plant device.

Controller device implements the control rules and interact with plant by receiving/sending data from sensors/to actuators. PLC consists of processor(s), memory, I/O and communication cards. It periodically scans the input variables, executes the control rules and updates the output variables according to a cycle period, called *Pplc*.

Plant devices are the sensors and actuators that enable respectively to collect information about the plant status and to operate actions on the plant process. Considering WNDCS, sensors and actuators are supposed to exchange information with control devices using remote I/O communicating through wireless network.

Manuscript received June 1, 2010.

Authors are with the Centre de Recherche en Automatique de Nancy (CRAN), UMR 7039 Nancy Université - CNRS, Campus scientifique, BP 70239, 54506 Vandoeuvre-lès-Nancy, France e-mail: {gilbert.habib, jean-francois.petin, thierry.divoux}@cran.uhp-nancy.fr).

Network device ensures the exchange of information between the *plant* and *control devices*. In the WNDCS this paper is focusing on, wireless communication follows the IEEE 802.11 standard, also called Wi-Fi. All the devices of the WNDCS are equipped with communication card that allow sending information according to a cyclic period, called *Pcard*, and receiving information that are immediately processed.

In general, three layers are considered in WNDCS: Physical, MAC and Application layers. The Physical layer is responsible for transmitting and receiving signals. Many technologies can be chosen in wireless network example: 802.11a, b, and g. The MAC layer manages the access to the medium using CSMA/CA (Collision Avoidance) algorithm. A station that wants to send a message is listening for the medium during a Distributed Inter-Frame Space (DIFS) time. If the medium is busy, a random waiting time, called *backoff time*, is taken. This time is between 0 and a variable called Contention Window (CW), where CW is between CW_{min} and CW_{max}. The station begins to decrement backoff time as long as the medium is idle and it will be frozen when the medium becomes busy. When *backoff time* becomes equal to zero, the station sends immediately the packet. Transmission of a message may fail because of noise that can affect physical layer or because of collisions that can occur if two same backoff times are taken for two different stations by the MAC layer. The *Application layer* manages the information sending and receiving procedures: periodical encapsulation of information to be sent into packets according to predefined parameters (packet size, sampling period *Pcard* ...) and transmission of packets to lower layers.

B. Impact of the network behaviour on the WNDCS

Introducing IEEE 802.11 network in a discrete control system may have a negative impact on the predictability of its global behaviour. Indeed, non determinism of backoff procedure of CSMA/CA leads to jitter phenomena due to a random end-to-end delay between source and destination. Moreover, packet loss due to noise or collision may aggravate this problem because it requires an unknown number (but limited to a maximum value) of packet retransmission that leads to more uncertainty about the end-to-end delay or, in the worst case, information that is never received.

For continuous control system, this problem may disturb the stability of the control loop [2]. For DCS, it may lead to Response Time (ResT) that is not compliant with the system requirements in terms of temporal performance or safety constraints. Indeed, real-time discrete control applications often requires a limited time for the reaction to a stimulus input.

Response Time is defined by [3] to reflect the performance of a NCS which represents the roundtrip time from client to server and back to client including a processing time in the server. Also it can be defined as the

delay between the occurrence of an event in a sensor (i.e.: detecting the presence of an obstacle) and the reception by the actuators of the control reaction to this event (i.e.: order to stop the move of a device). This delay takes into account the time for this event to be received by the controller, the time required for control processing and the time for the control output to be sent to the actuators.

III. IMPROVING WNDCS PERFORMANCE

Many researchers have studied the influence of network parameters on the response time for non deterministic communication [4] and more particularly for IEEE 802.11 wireless communication [5][6]. If these works provide influence parameters that impact the response time, they do not suggest ways to improve the situation. Three main approaches have been proposed.

Control oriented approaches take into account the impact of communication during the control design by including estimation or prediction of the network behaviour into the control models [7][8]; main drawback of these approaches is the rough estimate of the network.

Network oriented approaches aims at optimising network quality of service [9] by tuning its controllable parameters (packet size, routings, topology, data rate ...); main drawback of these approaches is that the control requirements are not really taken into account.

At last *Control/Network oriented* approaches propose co-design approaches that integrate both the control requirements and models and the network quality of service optimisation [10][11]; these approaches introduce efficient ways to adapt the network with regards to the application needs but suffer from a high level of complexity required for a detailed modelling of control and network behaviour.

Our work locates in the *Control/Network oriented* approach. It is based on communication priorities tuning provided by the standard IEEE 802.11e. Indeed, IEEE 802.11e extension of IEEE 802.11 enables giving priority to each traffic of the network. More precisely, 802.11e includes a mandatory EDCA (Enhanced Distributed Channel Access) MAC mechanism. It provides four levels of priority from 0 (lowest priority) to 3 (highest priority). Every level has its own buffer in the MAC layer managed by standalone CSMA/CA. Each buffer, i.e each priority, has its own *backoff time* and DIFS (called AIFS in IEEE802.11e) parameters. The more the priority of traffic is higher, less the *backoff time* and the AIFS are small. Thus, the traffic will have more possibility to access the medium if it has a higher priority. Note that another mechanism available in IEEE802.11e, called HCCA (HCF controlled channel access (HCCA)) provides deterministic management of the medium access without using CSMA/CA. Unfortunately, HCCA is few implemented by commercial devices due to its complexity. Consequently, EDCA has been selected for our study.

Theoretically, the priority between the traffics in 802.11e must give a better performance compared to 802.11. Some

previous studies [12][13] do not show that when considering static and predefined priorities. Their simulations prove that the traffics with higher priorities will dominate the medium, thus this will penalize the traffic with less priorities. Therefore, the priority may play a negative effect on some traffic in the network. Classical recommendation to avoid this negative effect is to avoid using 0 priority. Our proposal is to promote a dynamic allocation of the priorities based on the estimated needs for the control application.

IV. PRIORITY MANAGEMENT ALGORITHM

In order to improve the WNDSCS performances in terms of security and velocity, an algorithm for adapting the network with regards to control needs is proposed. This algorithm is based on a dynamic variation of the traffic priorities to ensure fairness between the different stations according to the control application needs. In other words, the controller will define dynamic classification of its input information with regards to its importance for its current state.

The control system is assumed to be modelled using Finite State Automata defined by $(X, \Sigma, \delta, x_0, x_f)$ where X is a non-empty set of states, Σ is the alphabet, δ is the state-transition function: $\delta: X \times \Sigma \rightarrow X$, x_0 is an initial state, an element of X and, x_f is the set of final states ($x_f \subseteq X$).

Three sets of devices are thus defined with regards to the information criticism they send.

The controller selects the devices that will send *Critical information* that the controller is waiting for. It corresponds to the critical information for current control decision making; therefore, this information must arrive on time.

The controller selects the devices that will send information which are not important to take current decision but that will be useful for next step decision making; so it must be monitored, we call it *Normal information*.

At least, controller selects the devices that send information that it should not used in a near future, it is called *Unnecessary information*.

More formally, the definition of those three sets is based on the current state x_{cur} of the control model. A sub-set Σ_{plant} of the control model alphabet Σ is defined as the set of events coming from the sensors of the plant (the events are considered as controller inputs):

$$\Sigma_{plant} = \{s_1, s_2, \dots, s_m\} \subset \Sigma$$

Note that each sensor_{*i*} is assumed to be able to generate only one event s_i . Consequently, defining a set of event s_i is equivalent to define a set of sensor_{*i*}, (S_i).

Definition 1.a: C is defined as the set of events related to *Critical information*, i.e.:

$$C = \{s \in \Sigma_{plant} : \exists w \in (\Sigma \setminus \Sigma_{plant})^*, \delta(x_{cur}, ws) \text{ is defined}\}$$

where $(\Sigma \setminus \Sigma_{plant})^*$ represents all the words composed of events that belong to the set Σ but not to set Σ_{plant} .

Definition 1.b: N is defined as the set of events related to *Normal information*, i.e.:

$$N = \left\{ s' \in \Sigma_{plant} : \exists w, w' \in (\Sigma \setminus \Sigma_{plant})^*, s \in \Sigma_{plant}, \delta(x_{cur}, wsw's') \text{ is defined} \right\}$$

Definition 1.c: U is defined as the set of events related to *Unnecessary information*, i.e.:

$$U = \{s'', s''' \in \Sigma_{plant} \setminus (C \cup N)\}$$

The proposed algorithm behaves as following.

If the controller estimates that the network current quality of service may delay a *Critical Information* that may not arrive on time (and consequently generate a too late reaction), the priority of the device which will send this information must be increased. Evaluating the network current quality of service is done by monitoring the response time (ResT) of the devices and by comparing it to a theoretical value that is expected to ensure a normal behaviour of the control application. In other words, if the response time is upper than a threshold, the priority of the given device must be increased, otherwise the priority of the device is kept at its current level. Note that several methods for evaluating the response time are available: to keep the genericity of the proposed algorithm, the method we used will be further detailed in the case study section.

The priorities of the devices that send *Normal information* are kept in *default priority* value that has to be determined according to the use case (equal to 1 in most of the case).

The devices that send *Unnecessary information* are temporarily silenced. We consider that this action is equivalent to send a request for a priority equal to the value *Null* that will be interpreted by the device as a request for stopping transmissions.

To summarize, the proposed algorithm is implemented on the controller device within the control model section, that is run with a *Pplc* cycle (PLC sampling period) and involves: enumeration of C , N and U sets, estimation of the response time of sensors belonging to C and management of the priorities according to C , N and U sets and response time.

Algorithm – Priorities allocation

/* initialization */

$P(S_i) \leftarrow \text{default_priority}$

Threshold $\leftarrow \text{default_TH}$

For each Pplc sampling period do

Build C , N and U sets using traces from x_{cur} state

For each $s_i \in C$ do

Evaluate $\text{ResT}(S_i)$ (where S_i is the sensor emitting s_i and $\text{ResT}(S_i)$ is the response time of the sensor S_i);

If $\text{ResT}(S_i) > \text{Threshold}$ and $P(S_i) \leq 3$

Then $P(S_i) \leftarrow P(S_i) + 1$

Else $P(S_i) \leftarrow P(S_i)$

End_if

End_for

```

For each  $s_i \in N$  do
   $P(S_i) \leftarrow \text{default\_priority}$ 
End\_for

For each  $s_i \in U$  do
   $P(S_i) \leftarrow \text{Null /*Request to stop sending */}$ 
End\_for

Return  $P(S_i)$ 
End\_for

```

Note that this algorithm, and more specifically the definition of C , N and U sets, is based on the current state of the control model which is assumed to be described using finite state automaton. If the control model $\{A\}$ is described in a modular way with several coordinated automata $\{A_1, A_2 \dots A_n\}$, the following lemmas for computing the algorithm can be applied.

Lemma 1 : If $\Sigma_{A_i} \cap \Sigma_{A_j} = \emptyset$, i.e. A_i and A_j alphabets does not share common events, $C(A_i \| A_j) = C(A_i) \cup C(A_j)$, (resp. N and U), i.e. the enumeration of C (resp. N and U) will be equivalent if being computed from the synchronous product $A_i \| A_j$ than if obtained by merging C (resp. N and U) enumerations given by each automata.

Lemma 2: If $\Sigma_{A_i} \cap \Sigma_{\text{plant}} = \emptyset$, i.e. A_i alphabet does not contain sensor events, $C(A_i) = N(A_i) = U(A_i) = \emptyset$.

V. APPLICATION

A. Presentation of the case study

The proposed algorithm is experimented using a case study (Fig. 1) and through two simulation scenarios: with and without the proposed algorithm. The case study is composed of two cylinders (cylinder1 moves horizontally and cylinder2 vertically), their associated sensors (three sensors S1, S2, and S3 for cylinder1, two sensors S4 and S5 for cylinder2) and a PLC that control the system. We assume that each device (sensors, actuators and PLC) send and receive information using wireless I/O cards. This configuration is clearly unrealistic for this kind of application but is interesting for communication experiments and is representative of what could be industrial application in a near future when using wireless sensor networks.

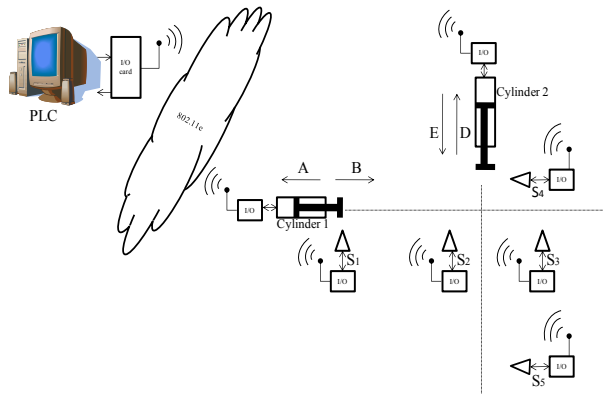


Fig. 1. Architecture of the WNDCS in the case study

Temporal performance of the WNDCS is required for avoiding collision between the two cylinders that share a common area. The main communication is the end-to-end delay between occurrence of a sensor event and the receipt by an actuator of a reaction control event. Control is designed voluntarily to provoke a dangerous situation by asking the two cylinders to get out simultaneously. The control is then expected to stop cylinder 1 into intermediate position to avoid collision, to wait for cylinder 2 freeing the common zone and then to end the roundtrip. This cycle is continuously repeated until the end of simulation scenario.

Evaluating the temporal performance of the WNDCS is done using two criteria:

- number of collision that have not been avoided due to communication delay (*safety property*),
- number of roundtrips done by the two cylinders that characterise the control rate (*vivacity property*).

B. Modelling of the case study

To evaluate the algorithm, simulation scenarios using OPNET (<http://www.opnet.com>) are done. This tool is a network simulator, providing libraries for different protocol network and an editor for describing the behaviour of devices using EFSM (Extended Finite State Machine). This formalism has been shown as compliant with automata used in our models [6]. Our OPNET models are divided into three sections.

The *plant model* is an emulation model that represents the behaviour of the plant devices using EFSM and generates the sensors signals. Note that the actuators are modelled in a modular way as proposed by [14]. For our study, two main attributes of plant models are considered: *Sensor size* and *Spacing size sensors* which are respectively translated into time needed by the cylinder to cross the sensor and time needed for reaching a sensor from another sensor.

The *communication model* represents the behaviour of the network:

- *Physical* and *MAC* layers are compliant with IEEE 802.11e standard and implemented using OPNET library.
- For *Application layer*, a packet created by PLC encapsulates all the actuator orders (A, B, E, D in Fig. 2) and is broadcasted. The broadcasting method minimizes the number of packet send by the PLC station. The packet created by each sensor is only composed by the efficient information and unicasted to PLC. This model also includes the definition of the sampling period P_{card} .

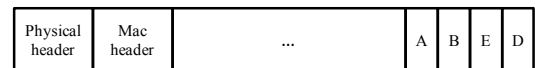


Fig. 2. Broadcasted packet format without proposed algorithm

The *PLC model* describes the control cycle (as presented in 5.1.) and is represented by modular automata (See Fig. 3):

- M1 and M2 estimate the state of the two cylinders according to sensor inputs and generate output variables

P_A_v1 and P_A_v2 according to this computed position,
- M3 manage the control cycle and collision avoidance according to cylinder1 and cylinder2 states as given by P_A_v1 and P_A_v2.

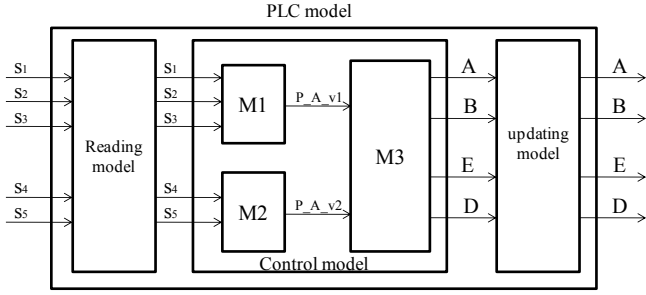


Fig. 3. PLC model

This basic control is enriched by the algorithm implementation that has to define priorities for the communication traffics. PLC station encapsulates in the broadcasted packet five additional fields (field 3 till 7), which represent the priorities recommended for each sensor (S1 till S5), see Fig. 4. These fields can take a value from 1 up to 3 that denotes the priority of the station or 0 if the station must be silenced.

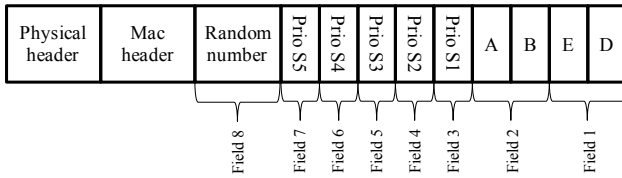


Fig. 4. Broadcasted packet format with proposed algorithm

As explained before, this algorithm is applied to the *control model*. It consists of selecting three sets: C , N and U which represent *Critical*, *Normal* and *Unnecessary* events respectively from Σ_{plant} . In this case study, $\Sigma_{\text{plant}} = \{s_1, s_2, s_3, s_4, s_5\}$.

Crossing from one state to another in M3 depends on the P_A_v1 and P_A_v2 variables, which are not included in Σ_{plant} . With regards to *lemma 2*, M3 generates empty sets for C_3 , N_3 and U_3 . The models M1 and M2 can be considered as local for each cylinder and its associated sensors. More precisely:

$$\Sigma_{\text{plant}} = \Sigma_{\text{plant} - M1} \cup \Sigma_{\text{plant} - M2}$$

where $\Sigma_{\text{plant} - M1} = \{s_1, s_2, s_3\}$ and $\Sigma_{\text{plant} - M2} = \{s_4, s_5\}$.

Consequently, alphabets of M1 and M2 have an empty intersection. According to *lemma 1*, the sets C , N and U can be respectively computed by merging respectively C_1 , N_1 and U_1 sets from M1 with respectively C_2 , N_2 and U_2 sets from M2.

Applying algorithm to control model is equivalent to apply separately the algorithm within M1 and M2 models.

For M1 and M2, parameters of the algorithm have been chosen as following:

- The threshold that is involved in the comparison with ResT has been chosen as *Sensor size*. This is justified by to provide the necessary time for stopping the cylinder before it leaves the sensor.

- Default priority has been chosen as equal to 1.

- Response Time (ResT) evaluation has been done according to the following procedure. The PLC encapsulates a flag as a random number (field 8 in Fig. 4) in the broadcasted packet. After recognising the flag, each sensor sends it back to the PLC within its information packet. The PLC station can then calculate the end-to-end communication time between sending the packet and receiving the response. Adding *Pplc* to this time give rise to response time (ResT).

At last, all this models are integrated into OPNET application using the following structure (Fig. 5): *plant* and *control* models are implemented into additional OPNET layers we introduced, *application layer* has been parameterised according to packet creation rules and sampling period we defined, and finally *physical* and *MAC* layers are implemented using OPNET libraries.

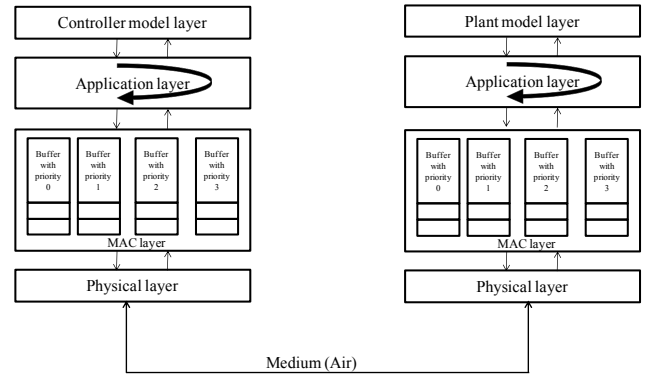


Fig. 5. OPNET model

C. Algorithm evaluation using simulation scenarios

After modelling the whole system in OPNET, we study the influence of the network attributes onto the system in two cases:

- without algorithm, where all traffics have the same priority equal to 1,
- with algorithm, where the priorities are computed in real time by taking into account communication quality of service and control state.

The static and dynamic parameters that are used by the simulation are given by Table 1. A Monte Carlo approach from 50 to 100 runs was taken for each simulation, depending on the gap between the current and mean values. Each simulation was launched for 200 seconds. As explained before, results of the simulation can be evaluated by considering the number of collisions between the two cylinders that have not been avoided due to communication delay (safety property) and, the number of roundtrips done by the two cylinders that characterise the control rate

(vivacity property). The first one must be minimized while the second one must be maximized.

Attributes	Values
<i>Pplc</i>	3ms
Packet size of the PLC packets	7bytes
Packet size of the sensors packets	5bytes
Sensors size	30ms
Spacing size sensors	1s
Data rate	11Mb/s
<i>Pcard</i> (ms)	From 1 to 25
Physical layer technology	802.11b

Table 1. Simulation parameters

Cross points of Fig. 6 and Fig. 7 represent simulations done without using the proposed algorithm: Fig. 6 represents the number of collision occurrences with respect to *Pcard* and Fig. 7 shows the numbers of roundtrips executed by the two cylinders. These curves can be decomposed into three sections:

- Simulations where $1 \leq Pcard \leq 4ms$ give rise to some collisions. This result can be explained by a lot of packet sent to the medium and the consecutive network over load meaning important delay and packet losses. Consequences on control application are bad estimation of the plant state, too late or inappropriate reaction from the control.

- Simulations where $5 \leq Pcard \leq 10ms$ does not present collisions. Indeed, for these values, the influence of the network on the global behaviour is minimized: the network induces small delay and few packet losses.

- Simulations where $11 \leq Pcard \leq 25ms$ give rise to many collisions. This result can be explained by the fact that *Pcard* is not compliant with real-time parameters of the plant. In this case, influence of network is very limited but some events to/from the plant may be not taken into account.

First section is important for WNDCS if we consider that some real-time control applications may require such time constraints. The resulting heavy load of the network may also be encountered for greater values of *Pcard* but with much more sending stations (in our case, only 6 stations are considered).

Fig. 6 and Fig. 7 (circle points) simulations are done using the proposed algorithm. For the simulations where $5 \leq Pcard \leq 25ms$, the results are quite the same with and without algorithm, due to the light influence of the network. For the simulations where $1 \leq Pcard \leq 4ms$, the algorithm improves significantly the results: no collisions are observed and a maximum number of roundtrips are obtained during the simulation. Nevertheless, one exception has to be mentioned. When *Pcard* is equal to 1ms, some collision happens. This is explained by the fact that *Pcard* (1ms) is near to the transmission time of a packet (approximately 0.8ms if we consider 802.11b-physical layer and ACK waiting time). In this case, the MAC layer buffers will fill rapidly and begins to drop packets coming from upper layer.

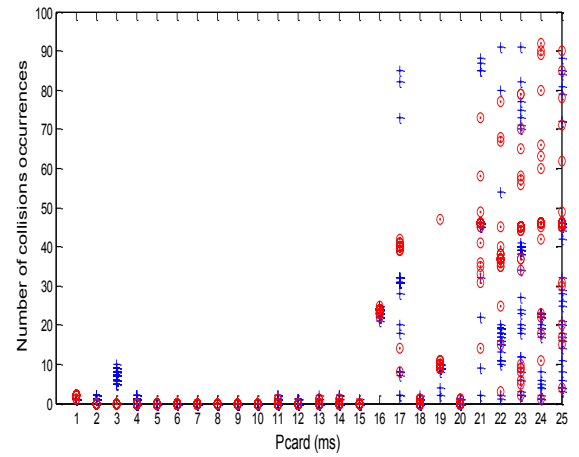


Fig. 6. Number of collisions occurrences without algorithm (cross points) and with algorithm (circle points)

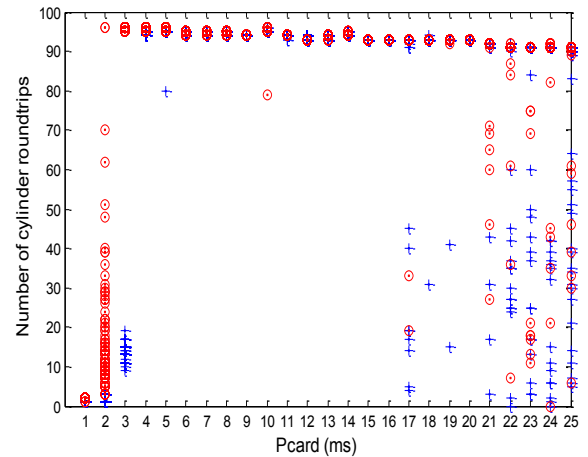


Fig. 7. Number of cylinder roundtrips without algorithm (cross points) and with algorithm (circle point)

VI. CONCLUSION

In this paper, the impact of a wireless communication network on the WNDCS performance in terms of safety and velocity is studied. To avoid degradations of control performance, due to collisions and packet losses in case of network heavy load, an algorithm for 802.11e priorities management has been proposed. The originality of this algorithm relies on a dynamic allocation of traffic priorities by taking into account the control requirements and current states of execution. Simulations of a cased study performed using the OPNET software tool have highlighted the efficiency of the proposed algorithm. These first simulation results should be reinforced by an implementation of the algorithm on laboratory-scale and industrial-scale platforms to make the proposed approach effective in practice.

REFERENCES

- [1] IEEE Std 802.11e, "IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements", Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.

- [2] J.P. Georges, T. Divoux, E. Rondeau, "Confronting the performances of a switched Ethernet network with industrial constraints by using the network calculus", in *International Journal of Communication Systems*, Vol. 18:877903, 2005.
- [3] E. Jasperneite, P. Neumann, "Switched Ethernet for factory communication", in *8th IEEE International Conference on Emerging Technologies and Factory Automation*, 2001, France
- [4] B. Denis, S. Ruel, "Measuring the impact of vertical integration on response times in Ethernet fieldbuses", in *12th IEEE Conference on Emerging Technologies and Factory Automation*, 2007, Greece.
- [5] J. Colandaira, G.W. Irwin, W.G. Scanlon, "Analysis of an IEEE 802.11b wireless networked control system" in *16th IFAC World Congress*, 2005, Prague, Czech Republic.
- [6] G. Habib, J.F. Pétin, T. Divoux, "Evaluation de l'influence d'un réseau de communication sans fil sur la commande d'un SED", in *Modélisation des Systèmes Reactifs*, 2009, France.
- [7] J. Nilsson, B. Bernhardsson, B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays", in *Automatica*, Vol. 34., 1998.
- [8] H. Chan, U. Özgüner, "Closed-loop control of systems over a communication network with queues", in *International Journal of Control*. Vol. 62., 1995.
- [9] D. Kim, D. Choi, P. Mohapatra, "Real-time scheduling method for networked discrete control systems" in *Control Engineering Practice*. Vol. 17, 2009.
- [10] L. Xiangheng, A. Goldsmith, "Wireless Network Design for Distributed Control", in *43rd IEEE Conference on Decision and Control*, Bahamas, 2004.
- [11] M.S. Branicky, S. Phillips S., W. Zhang, "Scheduling and Feedback Co-Design for Networked Control Systems", in *41st IEEE conference on decision and control*, Las Vegas, 2002.
- [12] A. Grilo, M. Nunes, "Performance evaluation of IEEE 802.11e", in *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Portugal, 2002.
- [13] C. Lu-ming, K. Bzunori, "Performance evaluation on IEEE 802.11e considering emergency calls in congested situation", in *The journal of china universities of posts and telecommunications*, 2007.
- [14] P. Marangé, "Synthèse et filtrage robuste de la commande pour des systèmes manufacturiers sûrs de fonctionnement", in *PhD thesis*, Université de Reims Champagne-Ardenne, 2008.