HAL
open science

Wireless communication networks and process control applications: Studying the influence of MAC protocols

Xuan Hung Nguyen, Guy Juanole, Gérard Mouney, Christophe Calmettes

To cite this version:

Xuan Hung Nguyen, Guy Juanole, Gérard Mouney, Christophe Calmettes. Wireless communication networks and process control applications: Studying the influence of MAC protocols. SACONAS 2010 - IEEE International Workshop Towards Smart Communications and Network technologies applied on Autonomous Systems (SaCoNAS), Dec 2010, Miami, United States. pp.1632 - 1637, 10.1109/GLOCOMW.2010.5700216 . hal-00656659

HAL Id: hal-00656659

https://hal.science/hal-00656659v1

Submitted on 4 Jan 2012

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.

Wireless communication networks and process control applications: studying the influence of MAC protocols

Xuan Hung Nguyen[1,2]
Guy Juanole[1,2]
Gérard Mouney[1,2]
[1]CNRS ; LAAS ; 7 avenue du Colonel Roche,
F-31077 Toulouse, FRANCE
[2]Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS :
F-31077 Toulouse, FRANCE
Email: xhnguyen@laas.fr, juanole@laas.fr, mouney@laas.fr

Christophe Calmettes
Université de Toulouse ; CUFR JF Champollion
Place de Verdun 81012 Albi Cedex 9, FRANCE
Email: christophe.calmettes@univ-jfc.fr

***Abstract*—The performances of the process control applications, which are implemented through networks, depend greatly on the scheduling of the messages. Furthermore, in the context of the wireless networks, because of the unreliable medium, it is difficult to meet the timing requirements imposed by the applications dynamics. The objective of this paper is, by considering a Wireless Local Area Network (WLAN) and a process control application, to compare the performances of this application by using MAC protocols of the type CSMA (which manage the messages scheduling): two protocols with collision avoidance (CA), provided by manufacturers (IEEE 802.11 - WiFi, IEEE 802.15.4 - ZigBee) which do not prevent the occurrence of collisions; a proposed free-collision protocol which is based on priorities specified by means of BlackBurst messages.**

I. Introduction

Wireless networks and in particular wireless LANs (WLANs) are more and more used to implement process control applications. In this context, the MAC protocol of the WLANs has the very important role to manage the message exchange scheduling.

The protocols, based on a CSMA/CA mechanism, are very used and one objective of this paper is to evaluate two simple protocols provided by manufacturers (IEEE 802.11 WiFi and IEEE 802.15.4 ZigBee). We consider here the DCF (Distributed Control Function) mode for WiFi and the unslotted CSMA/CA mechanism for ZigBee.

As these protocols induce collisions and as they don't have a deterministic behavior, they are not really destinated to real time applications and particularly process control applications, but it is interesting to evaluate them in such a context and to complement this analysis with the comparison with a deterministic protocol.

It is why in this paper we propose a CSMA protocol without collision which is based on priorities implemented by means of BlackBurst messages. Our work concerns the WLANs with the following assumptions: any station can send data to another in a single hop; each station is in the radio range of the others, *i.e.* they haven't the hidden terminal and exposed terminal problem.

This paper includes three parts: the first part presents the context of the study (the considered process control application, *i.e.* an inverted pendulum mounted on a trolley; a short presentation of the main ideas of the DCF mode for WiFi (noted in the following DCF-WiFi) and Unslotted CSMA/CA ZigBee (noted in the following U. ZigBee); the simulation which will be used for the study); the second part presents the protocol without collision which is proposed (several works [1], [2], [3], [4] have already presented such kinds of protocols but, in our mind, their presentation is not always very clear and we try here to make a pedagogical and understandable presentation of such a protocol); the third part presents the comparison of the results, obtained in terms of the Quality of Control (QoC) of the application, with each protocol used for the implementation of the inverted pendulum through a WLAN.

II. Context of the study

A. The considered process control application

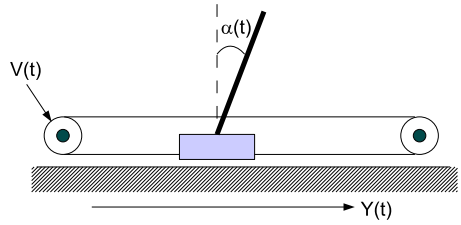

Fig. 1. Inverted pendulum on a trolley.

The process that we want to control through a network is an inverted pendulum mounted on a trolley. The inverted pendulum can move horizontally thanks to a DC motor. A schematic diagram of the system is given in the figure 1. The

state $Y(t)$ corresponds to the linear position of the trolley on the horizontal axis. The pendulum angle from the vertical is denoted $\alpha(t)$. The control signal $V(t)$ of the process is the voltage applied to the DC motor. The objective is to move the pendulum from a position $Y = Y_0$ to a position $Y = Y_{ref}$ while keeping the pendulum vertical $\alpha(t) = 0$. The state vector $X(t)$ is $X(t) = \left[\ Y(t)\ \ \alpha(t)\ \ \dot{Y}(t)\ \ \dot{\alpha}(t)\ \right]^T$.
The control law is a discrete time state feed-back which leads, with a well choosen sampling period $h = 60$ ms, to the desired specification: rise time $tr \simeq 400$ ms and damping ratio $\xi \simeq 0.7$.

B. Simulation of the implementation of the "Inverted Pendulum on a trolley" through a network

We use the simulator TrueTime [5], a toolbox based on Matlab/Simulink which allows to simulate both networks mechanisms (message scheduling, communication protocols ...) and control systems (modelled by transfer function, state space ...).
The implementation is based on three tasks: a sensor task (time-triggered), a controller task and an actuator task which are event-triggered. The sensor task sends periodically (period of 60 ms) a frame, including the state vector $X(t)$, to the controller task. The frames, going from the sensor task to the controller task, form the flow fsc. The controller task, after the reception of a frame of fsc, computes the control signal and sends it in a frame to the actuator task. The frames, going from the controller task to the actuator task, form the flow fca.

C. DCF-WiFi and U. CSMA/CA ZigBee

We present briefly the main ideas. When a station has a message to transmit (all the details can be found respectively in [6], [7]):

- DCF-WiFi: at first there is a test of the medium and if the medium is found free for at least a DIFS duration (DCF Interframe Space) the message is sent. Otherwise, we have a Backoff based on a random delay implemented with a timer; the timer is decremented when the medium is free. When the timer reaches the value 0, the message is sent. If a collision is detected, a new Backoff time is chosen. When the receiver gets the message, it sends, after a time SIFS (Short Interframe Space), an ACK which allows the sender to conclude the non existence of collision. Note furthermore that after a successful transmission, we have what is called the post-backoff mechanism to ensure that there is always a random time between two successive messages (note still that, if after the post-backoff, the medium stays free, we do not need to make a DIFS before the next transmission).
- U. ZigBee: Here (fundamental difference with DCF-WiFi) we wait for a random duration without testing the medium (Backoff period) and at the end of this duration, we test the medium (Clear Channel Assessment (CCA)); if the medium is free, we transmit the message; otherwise we wait again for a new Backoff period. When the receiver gets the message, it sends an ACK which allows the sender to conclude that there is no collision.

D. Criterion used in the simulation

The simulations are performed on a time horizon of 4 seconds for a set point Y_{ref}. This set point is a position echelon type which has an amplitude 0.1 applied at the time $t = 0.5s$. To evaluate the QoC of this system, we will use a performance criterion based on the variation of α:

$$J = \sum_{k=1}^{n} \alpha_k^2$$

where n is the number of periods during which the angular deviation is calculated. Higher is J, more degraded is the QoC. The degradation of this criterion is written: $\frac{(J-J_0)}{J_0} = \frac{\Delta_J}{J_0}$ with J_0 performance criterion of the inverted pendulum system in the case without network (we have $J_0 = 2.7762 \times 10^{-3}$). The figure 2 represents the time response of this application without the network: the position Y (m) and the angle α (rad). The performances without the network are taken as reference for the analysis with the differents protocols.

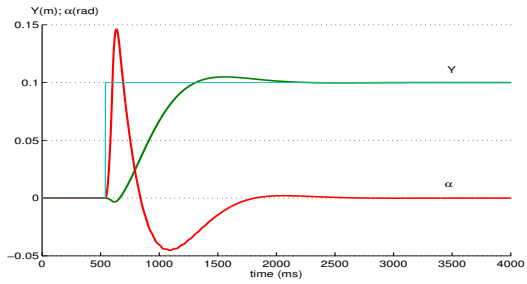

Fig. 2. Time response - without network.

III. Proposition of a collision-free MAC protocol

A. General idea of the protocol

This protocol is based on a stage of announcement messages before the stage of the data sending. An announcement message is made up of a sequence of energy pulses [2] (an energy pulse is called a BlackBurst (BB) and an announcement message is still called a BB message). The length of a BB message is proportional to the priority of the data to send. There are many ways to express the priority. Here we consider that the priority is given by the data flow IDentifier (ID) and is unique (it is a static priority). When nodes want to transmit, they first try to send their announcement messages and it is the node with the longest announcement (i.e. the highest priority) which wins the competition and then it can transmit a data frame.

This announcement mechanism is characterized by three phases: first, an OBServation phase of the free medium (called OBS1); second, the sending of the announcement message (called announcement phase), third, a new OBServation phase of the medium (called OBS2) which allows, for a node i, to determine the result of the announcement phase, *i.e*:

- If the medium is free, that means that the node i has the longest BlackBurst, *i.e.* the highest priority (then it is the winner) and it can transmit a data frame.
- If the medium is busy, that means that there is at least one node which has a priority higher than the one of the node i. The node i loses the competition and it must wait for the end of the data transfer done by the winner. Then it will restart the announcement mechanism in order to try to access to the medium.

We call TBB, TOBS1 and TOBS2 the durations of respectively an energy pulse BB, OBS1 and OBS2. These durations are the fundamental elements of the protocol.

B. Evaluation of TBB

The evaluation of TBB requires to specify the consequence in a CSMA context of the ambiguity of the expression "free medium at a time t at a station i" because this view is a local view whereas the medium is geographically distributed and then the local view can be different from the global view. We characterize this consequence by introducing the concept of "ambiguity temporal window" which represents the maximal possible duration between the sending of a message at a station and the occurence of a collision on this message.
From the consideration of the "ambiguity temporal window", we will deduce the constraints on TBB in order to have no more collision but only a result "winner- looser(s)" for a competition to access the medium.

1) Determination of the ambiguity temporal window: Call τ_{ST} and τ_{TT} respectively the Sensing Time and the Turnaround Time in each station and τ_{PT} the propagation time between the two most remote stations (maximal propagation time). Consider the figure 3 which wants to represent from a theoritical point of view the "ambiguity temporal window". In a station i we have a Request to Send at the time $t - \tau_{ST}$ and we suppose that the medium stays free during τ_{ST}. Then the medium is detected free at the time t and the station i decides to send a message. After the Turnaround Time τ_{TT} the message is sent and the beginning of this message arrives at the remote station j just at the time $(t + \tau_{TT} + \tau_{PT})$. If the station j is just finishing to test the medium at the time $(t + \tau_{TT} + \tau_{PT})$ (after a Request to Send at $(t + \tau_{TT} + \tau_{PT} - \tau_{ST})$), as this station saw the medium free during τ_{ST}, it will send a message and then there will be a collision with the message arriving at the station j. The duration $(\tau_{TT} + \tau_{PT})$ represents the "ambiguity temporal window".

Now if we want to consider the problem of the "ambiguity temporal window" from an implementation point of view, it is said [8] that, if the sensing time τ_{ST} is partially busy (the energy integral during τ_{ST} is lower than a threshold) the medium is considered free. That means that, in the station j, if the sensing time overlaps the instant $(t + \tau_{TT} + \tau_{PT})$, the medium can still be considered as free by the station j. Then the "ambiguity temporal window" has a value larger than the theoritical value $(\tau_{TT} + \tau_{PT})$. To characterize the

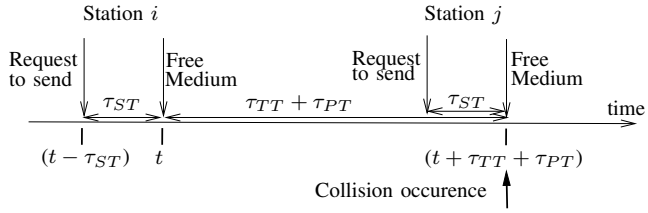

Fig. 3. Theoritical "ambiguity temporal window"

implementation aspect, we consider here a worst case *i.e.* we add τ_{ST} to the theoritical value of the "ambiguity temporal window" and in the following we consider for the "ambiguity temporal window" the value $(\tau_{TT} + \tau_{PT} + \tau_{ST})$.

2) Scenarios for the evaluation of TBB: Suppose a station i which finds the medium free at the time t and decides to send a message. In the context of strict CSMA and the "ambiguity temporal window", we can have two kinds of collision situations which will occur: 1/ a collision, at the start of its sending, in the station i which results from a message sent by a station j at the time $t - (\tau_{TT} + \tau_{PT} + \tau_{ST})$ and which arrives at the time t in the station i; 2/ a collision (at the start of its arrival) in a station j which results from a message sent by this station j at the time $t + (\tau_{TT} + \tau_{PT} + \tau_{ST})$.
We now have to find the constraints on the TBB values which will transform these situations in a situation "winner-looser(s)" and then we have to consider the following scenarios:

- a station i which has the priority $(k+1)$, k being a positive integer (*i.e.* the BB message has a length of $(k+1)$TBB), sends, at the time t, its BB message,
- a station j, which has a priority immediately lower than the priority of the station i (*i.e.* a priority k and then the BB message has a length of kTBB), sends its BB message either at the time $t - (\tau_{TT} + \tau_{PT} + \tau_{ST})$ or at the time $t + (\tau_{TT} + \tau_{PT} + \tau_{ST})$.

The scenario, which expresses the hardest constraints, is when the station j sends later than the station i, *i.e.* at the time $(t + (\tau_{TT} + \tau_{PT} + \tau_{ST}))$, because the BB message from the station j has more chance to be able to go beyond the BB message of the station i. For space reasons in the paper, we express the constraints only in this scenario.

3) Constraints on TBB: In order to be sure that the station i is always the winner, we have to find the conditions such that:

- Case 1: the propagation in the station i of the BB message coming from the station j does not exceed the BB message generated by the station i: $t + (\tau_{TT} + \tau_{ST} + \tau_{PT}) + \tau_{TT} + kTBB + \tau_{PT} \leq t + \tau_{TT} + (k+1)TBB$ true if TBB $\geq 2\tau_{PT} + (\tau_{TT} + \tau_{ST})$.
- Case 2: the propagation in the station j of the BB message coming from the station i always exceeds the BB message generated by the station j: $t+\tau_{TT}+(k+1)TBB+\tau_{PT} > t + (\tau_{TT} + \tau_{PT} + \tau_{ST}) + \tau_{TT} + kTBB$
  true if $TBB > (\tau_{TT} + \tau_{ST})$.

The condition of the case 1 defines the constraint on TBB: $TBB \geq 2\tau_{PT} + (\tau_{TT} + \tau_{ST})$.

Remark: Note after this strict demonstration that we find a constraint on TBB different of that given (but not demonstrated) by [1] and which is: TBB $\geq 2(\tau_{PT} + \tau_{TT} + \tau_{ST})$.

C. Evaluation of TOBS1 and TOBS2

First we evaluate TOBS2: TOBS2 must be equal to TBB. Indeed, a station after having sent its BB message, must see if there exists a BB message with a higher priority (then, with at least, an energy pulse in more, i.e. duration TBB).
TOBS1 can only be specified when TOBS2 has been specified. The role of TOBS1 is to avoid an intrusion during an exchange in progress and then we must take TOBS1 $>$ TOBS2.

D. An example of BB message and data frame exchanges

The principle of this access mechanism is illustrated in the figure 4 where we consider exchanges between three stations A, B, C and the station A is transmitting a Data frame at the beginning of the considered scenario. Nodes B and C want to send a Data frame at the moments TR (Transmission Request). They have to wait until the end of the current transmission of the node A and then they wait for TOBS1 (i.e. phase 1: observation). After TOBS1, nodes B,C send the announcement messages (i.e. announcement phase). At the end of transmission of BB messages, node C tests the medium during TOBS2, node C finds the medium occupied and so it defers its transmission; node B, after TOBS2, finds the medium free and it sends its Data frame. During the transmission of the station B, the station A has again a need to send a Data frame (request TR). This Data frame has the priority lower than the priority of the Data frame of the station C which explains the right part of the figure.

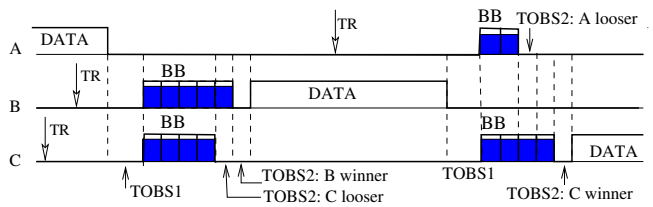

Fig. 4. Message exchange of the proposed MAC protocol.

E. The implementation of the BB messages

Suppose IDs of n bits and consider, like in the network CAN [9] that the highest priority is associated to the smallest ID: the BB message duration is $(2^n - \sum_{i=0}^{n-1} b_i 2^i)$TBB, b_i being the value of the bit of rank i. More simply, the BB duration of the message k is $(2^n - ID_k)$TBB. In the following, we give directly the priorities in terms of the length of the BB message (number of BBs). Furthermore, it is interesting to remark that, if we do not need all the combinations allowed by the field ID, it is better to choose the combination which corresponds to the highest values of ID (then the BB messages have the shortest lengths).

IV. MAC PROTOCOL COMPARISON

We compare four MAC protocols with respect to their ability to implement the process control application "Inverted pendulum on a trolley": the protocol DCF-WiFi; the protocol U. ZigBee; two protocols of the type "collision free", called BB-WiFi and BB-ZigBee, which use the same format of the Data frame and the same bit rate of respectively DCF-WiFi and U. ZigBee.
The application is characterized by five data, four data for the state vector (each state is represented by 16 bits) which are sent in the Data frames (then 64 bits) of the the flow fsc; the control signal (16 bits) which is sent in the Data frames of the flow fca.
The bit rates are 1 Mbit/s for DCF-WiFi and 250 Kbits/s for U. ZigBee.

A. Values of the parameters related to the different protocols

The values related to DCF-WiFi and U. ZigBee (noted U. ZigBee) are in the table I.

	Data frame fsc bits	Data frame fca bits	ACK frame bits	DIFS μs	SIFS μs	τ_{TT} μs	τ_{ST} μs
DCF-WiFi	528	480	304	50	10	5	15
U. ZigBee	184	136	88	///	192	192	128

TABLE I
PARAMETERS OF DCF-WIFI AND U. ZIGBEE

Concerning BB-WiFi and BB-ZigBee, we give on the table II the value of ID field length, TBB, TOBS1, TOBS2. To satisfy TOBS1$>$ TOBS2, we choose the smallest difference which is given by the value SIFS in each network. We have: TOBS1 = TOBS2 + SIFS.
Remark: With these values, with all the different protocols, during the sampling period h, we have the necessary time to do the exchanges of a frame fsc and fca.

	ID length	TBB	TOBS1	TOBS2
BB-WiFi	6 bits	20 μs	30 μs	20 μs
BB-ZigBee	6 bits	320 μs	512 μs	320 μs

TABLE II
VALUES OF ID LENGTH, TBB, TOBS1 AND TOBS2.

B. Considering a network dedicated to one "Inverted pendulum on a trolley"

The goal of this subsection is to show the influence of the protocol structure i.e. the data frames and all the overhead required for the exchanges.

We consider all the studied protocols DCF-WiFi, U. ZigBee, BB-WiFi and BB-Zigbee. Concerning these two last protocols, we want to show, in particular, the influence of the implementation of the priorities: we consider two cases, the case 1 which gives the shortest BB messages (the ID of a frame of the flow fca is 62 and the ID of a frame of the flow fsc is 63 which gives BB message lengths of respectively 2TBB and 1TBB)

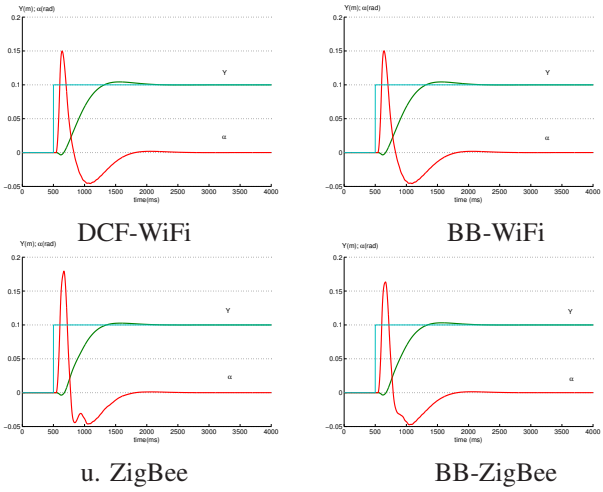

Fig. 5. Process control time responses.

and the case 2 which gives the longest BB messages (the ID of a frame of the flows fca and fsc are respectively 0 and 1 which give BB message lengths of respectively 64TBB and 63TBB).

We present the results of the performance criterion, i.e. QoC, on the table III (the cases 1 and 2, relative to BB-WiFi and BB-ZigBee, are noted with respectively (*) and (**)).

We can do the following comments:

- DCF-WiFi gives better results than U. ZigBee (that results from the low bit rate of ZigBee (four times lower) and also the Backoff at the beginning of each access).
- BB-WiFi and BB-ZigBee are sensitive to the choice of the ID values.
- When we choose ID values which give the shortest BB message lengths, BB-WiFi has performances which are equal to the DCF-WiFi performances.
- BB-ZigBee improves U. ZigBee when we have the shortest BB message lengths; on the other hand, when we have the highest BB message lengths, BB-ZigBee becomes very wrong (it is because the durations τ_{TT} and τ_{ST} are very long).

	DCF-WiFi	BB-WiFi	U. ZigBee	BB-ZigBee
$\Delta J/J_0\%$	6.04	(*) 6.55 (**) 26.6	43.4	(*) 24.3 (**) 7.5e+27

TABLE III
QoC.

We have represented in the figure 5, by considering for BB-WiFi and BB-ZigBee the case 1 of the priority implementation, the time responses of the process control application. We can see (by comparing with the reference of the figure 2) the good performances of WiFi and BB-WiFi. We see also the improvement of BB-ZigBee with respect to U. ZigBee (smaller overshoot and curve with a better shape).

$fca1$	$fca2$	$fca3$	$fca4$	$fsc1$	$fsc2$	$fsc3$	$fsc4$
8	7	6	5	4	3	2	1

TABLE IV
BB MESSAGE LENGTHS.

C. Considering a network shared by four identical "Inverted pendulum on a trolley" applications

We consider only the protocols DCF-WiFi and BB-WiFi (case 1 which gives the best results in the previous subsection) and we want to see here the difference in terms of QoC in a context of network sharing.

We call P1, P2, P3, P4 the four process control applications and we consider the priority scheme for the messages of the flows $fcai$ and $fsci$ (i = 1, 2, 3, 4) which is represented on the table IV in terms of the BB message lengths. This choice expresses that we consider in terms of priorities between the process control applications: priority P1 $>$ priority P2 $>$ priority P3 $>$ priority P4.

We consider also that the sensors are synchronous (they produce informations at the same time). It is the worst case in terms of competition.

The results in terms of QoC (table V and figure 6) show with respect to the table III (only one process control application on the network) a degradation of the QoC (which is normal) but the degradation is, with WiFi, similar ($\approx$ 25%) for all the applications whereas the degradation, with BB-WiFi, depends on the priority of the applications (the application P1 has a very small degradation).

$\Delta J/J_0\%$	P1	P2	P3	P4
DCF-WiFi	26.9	24.3	28.5	28.7
BB-WiFi	7.7	17.3	29.5	45.5

TABLE V
QoC.

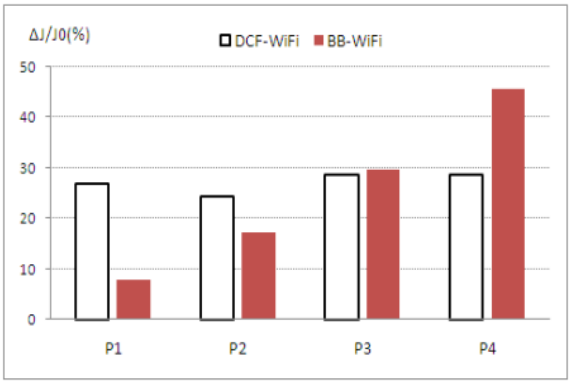

Fig. 6. QoC for DCF-WiFi and BB-WiFi.

The explanation of these results is given by the observation of the exchanges (Fig. 7 and 8) in the network with TrueTime. With DCF-WiFi, we have the random access (Fig. 7) and then we cannot favor one application; with BB-WiFi, we have the

deterministic access (Fig. 8) which depends on the priority and then we can discriminate the applications.

The figure 8, which shows very well the deterministic behavior, needs some explanations:

- at the time 0, all the sensors want to access to the medium and then only $fsc1$ (higher priority for a sensor flow) accesses the medium,
- then the flow $fca1$ accesses the medium; note that when $fca1$ accesses the medium, all the flows $fsc2$, $fsc3$ and $fsc4$ try to access to the medium but cannot access because their priorities are lower than the priority of the flow $fca1$ (the black thin line in the curves of $fsc2$, $fsc3$ and $fsc4$ represents the BlackBurst messages which have lost the competition).
- then all the sensors, after the access of the flow $fca1$ try to access the medium; only $fsc2$ accesses (the curves of the flows $fsc3$ and $fsc4$ have still a black thin line which represents that their BlackBurst messages have lost the competition).
- ... and so on.

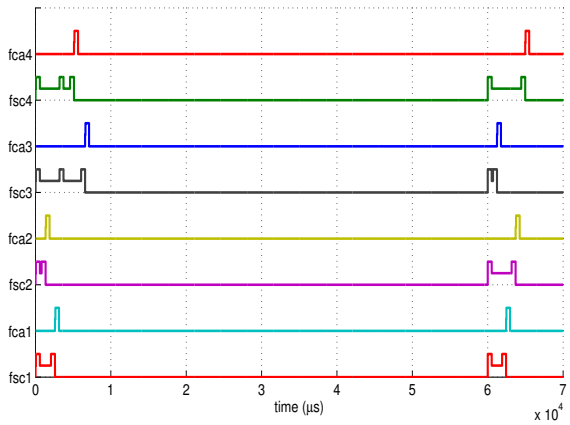

Fig. 7. DCF-WiFi: BlackBurst and Data message exchanges.

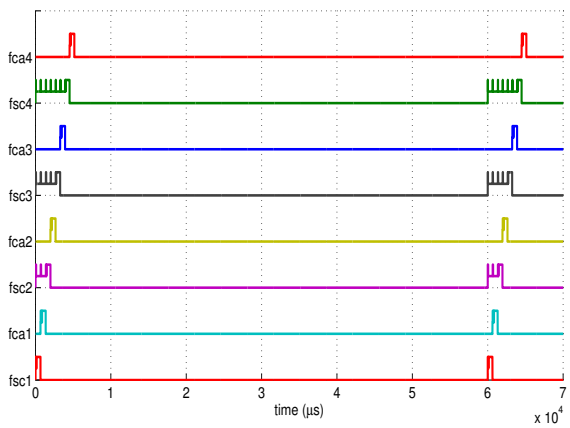

Fig. 8. BB-WiFi: BlackBurst and Data message exchanges.

V. Conclusion and Perspectives

This work presents, in our opinion, two contributions:

- The first one concerns a rigorous and educational presentation of the free collision protocol with static priorities based on BlackBurst messages (we explain and demonstrate the constraints on the BlackBurst durations by considering different scenarios).
- The second one concerns the application of this protocol to the implementation of a process control through a network; we have considered the frames of WiFi and ZigBee which is a pessimistic case (the frames are very long); we have also compared this protocol to the protocols DCF-WiFi and U. ZigBee giving interesting results; the drawback of this protocol is the length of the BlackBurst messages which can become prohibitive if we have many priorities.

Prospective works are in three directions:

- To consider hybrid priorities (dynamic part and static part) which can be very interesting in a context of process control applications on wireless network.
- To implement the priorities in the frame exchanges by using a mixing of BlackBursts and WhiteBursts (WhiteBursts are waiting times).
- To reduce the lengths of the priority messages by trying to implement the CAN technique in wireless context (we know that some works [10], [11] have been done in this direction but these works need still to be improved).

References

[1] J. Sobrinho and A. Krishnakumar, "Real-time traffic over the ieee 802.11 medium access control layer," *Bell Labs Technical Journal*, vol. 10, no. 1, pp. 172–187, 1996.

[2] ——, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1353–1368, Aug 1999.

[3] S. Kumar, V. S. Raghavan, and D. Jing, "Medium access control protocols for ad hoc wireless networks: A survey," *Ad Hoc Networks*, vol. 4, pp. 326–358, May 2006.

[4] A. Pal, A. Dogan, and F. Özgüner, "Mac layer protocols for real-time traffic in ad-hoc wireless networks," in *ICPP '02: Proceedings of the 2002 International Conference on Parallel Processing (ICPP'02)*. Washington, DC, USA: IEEE Computer Society, 2002, p. 539.

[5] M. Ohlin, D. Henriksson, and A. Cervin, *TrueTime 1.5 – Reference Manual*, Lund Institute of Technology, Sweden, January 2007.

[6] B. Crow, W. I., K. L.G., and S. P.T., "IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 35, pp. 116–126, sept 1997.

[7] I. S. S. Board, *Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)*, 2003.

[8] A. Kiryushin, A. Sadkov, and A. Mainwaring, "Real-world performance of clear channel assessment in 802.15.4 wireless sensor networks," *Sensor Technologies and Applications, International Conference on*, vol. 0, pp. 625–630, 2008.

[9] Bosch GmbH, *CAN specification 2.0 (A)*. www.semiconductors.bosch.de/pdf/can2spec.pdf, 1991.

[10] T. You, C.-H. Yeh, and H. Hassanein, "A new class of collision prevention mac protocols for wireless ad hoc networks," in *Proceedings of IEEE International Conference on Communications (ICC'03)*, 2003, pp. 1135–1140.

[11] N. Pereira, B. Andersson, and E. Tovar, "Widom: A dominance protocol for wireless medium access," *Industrial Informatics, IEEE Transactions on*, vol. 3, no. 2, pp. 120–130, May 2007.