



Specification of Collision-Free CSMA MAC Protocols for Wireless LANs

Guy Juanole, Xuan Hung Nguyen, Gérard Mouney

► To cite this version:

Guy Juanole, Xuan Hung Nguyen, Gérard Mouney. Specification of Collision-Free CSMA MAC Protocols for Wireless LANs. 2013. hal-00668745v4

HAL Id: hal-00668745

<https://hal.science/hal-00668745v4>

Submitted on 24 May 2013

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.

Specification of Collision-Free CSMA MAC Protocols for Wireless LANs

Guy Juanole^{1,2}, Xuan Hung Nguyen^{1,2} and Gérard Mouney^{1,2}

¹*CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France*

²*Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France*
{juanole, xhnguyen, mouney}@laas.fr

Keywords: Medium Access Control (MAC) protocols, CSMA MAC protocols, collision free MAC protocols, BlackBurst technique, CANlike protocol.

Abstract: Collision-free Medium Access Control (MAC) protocols based on Carrier Sense Multiple Access (CSMA) are interesting solutions in considering real-time traffic in a wireless context. The goal of this paper is to precisely make an exhaustive presentation (which we think to be understandable and pedagogical) of the main collision-free CSMA MAC protocols for single-hop Wireless Local Area Networks (WLANs), considering priorities associated to the frames. The priorities are represented by either the BlackBurst technique or an adaptation of the wired CAN bus protocol to the wireless context that we call CANlike. The cornerstone of this work is the concept of Ambiguity Time Window that we have defined which allows to specify the protocol parameters of the considered protocols. In our knowledge, such kind of study has never been done.

1 INTRODUCTION

Wireless networks and, more particularly, WLANs are more and more used today in the industrial area where we have real-time applications which require Quality of Service (QoS) guarantees. In this context, the MAC protocols which implement the scheduling of the frames on a shared radio channel have an essential role.

Two main types of MAC protocols are TDMA (Time Division Multiple Access) and CSMA (Carrier Sense Multiple Access). CSMA is a totally distributed procedure whereas TDMA requires some centralized schemes. Then, CSMA is more flexible than TDMA with respect to changes in a network (adding or withdrawing stations and/or applications). Furthermore, CSMA is more suitable for sporadic traffic. In this work, we consider MAC protocols of the type CSMA for single-channel and single-hop WLANs (i.e. each node is in the transmission range of the other nodes, we do not have the hidden terminal and exposed terminal problem).

The main MAC protocol used in WLANs and based on CSMA is IEEE802.11 DCF (Crow et al., 1997) (DIFS, Backoff, CW (Contention Window)) which does not support packet priorities and traffic differentiations. Another main one supporting packet priorities is IEEE 802.11e EDCA (802.11e, 2005) (AIFSSs, Backoff, CW) which allows traffic differenti-

ations (by means of different AIFSSs which expresses priorities represented by different time-outs). Some others in which the priorities are defined based on different sizes of Inter frame spaces, CW and Backoff are (Deng and Chang, 1999), (Mangold et al., 2003), (Vaidya et al., 2005). However, the big drawback of these protocols is that collisions can always occur due to the asynchronism between the transmission needs and the random behavior of the Backoff mechanism.

Obtaining collision-free CSMA MAC protocols and the QoS guarantees is possible by associating priorities represented by messages preceding the frames. The first approach is to use the BlackBurst technique (Sobrinho and Krishnakumar, 1996), (Pal et al., 2002). The idea is to let contending nodes send first jamming signals (called BlackBurst (BB) messages) of length according to the priority. The node that has the longest signal (i.e. the highest priority) wins the competition and then sends its frame. The drawback of this technique is that if we have a great priority number, the jamming signals will be very long and give important time delays. The second approach is to adapt the MAC protocol of the wired CAN bus (Bosch GmbH, 1991) (the priority of the frame is expressed by the ID field which precedes the data field) to the wireless context (You et al., 2003), (Pereira et al., 2007). We call this protocol CANlike. In this study, we are interested in the collision-free CSMA MAC protocols and thus we consider these two ap-

proaches.

After a careful reading of the quoted references and many other papers, we think that these works do not clearly and rigorously state, at first, the specific problematic of CSMA (collision potentiality: why, when and how *i.e.* by considering and explaining the role of the physical parameters of the transceivers and the channel) and, then, the different mechanisms in order to define collision-free protocols. That is the reason why in this paper we want to give a pedagogical presentation on the collision-free CSMA MAC protocols in a wireless context.

This paper includes the following sections: the section 2 concerns preliminaries which are necessary to well understand and specify the complete problematic; the section 3 concerns the specification of MAC protocols based on BB messages; the section 4 presents the specification of the CANlike protocol; the section 5 concerns the conclusion.

2 PRELIMINARIES

We present knowledges concerning the physical (PHY) and MAC layers which are absolutely essential for the specification of collision-free CSMA MAC protocols.

2.1 Wireless Transceiver

In a wireless context (contrarily to the wired context), a transceiver cannot simultaneously send and receive on a channel and has three states: Transmitter, Receiver, Sleeper. Here we do not consider the state “sleeper” which is used for considerations of energy economy.

Two time attributes characterize the transceiver behavior: the channel Sensing Time τ_{ST} and the Turnaround Time τ_{TT} . τ_{ST} allows the transceiver to test the channel state (busy or idle): it is busy or idle depending on the detected energy on τ_{ST} which is higher or lower than a prefixed threshold (this represents the “Clear Channel Assessment” (CCA)). τ_{TT} is the time to go from the receiver (transmitter) state to the transmitter (receiver) state. During a CCA, if the channel is detected busy, the transceiver still stays in the receiver state in order to continue listening to the channel; on the other hand, if the channel is detected idle, the transceiver (after a τ_{TT}) goes in the transmitter state which allows the MAC entity to send a frame. After the frame transmission, the transceiver comes back to the receiver state.

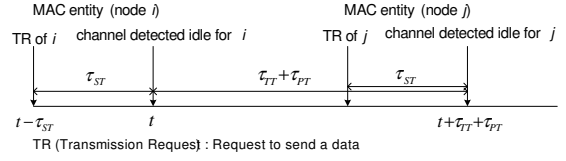


Figure 1: Ambiguity Time Window.

2.2 Concept of “Ambiguity Time Window”

We introduce this concept in order to quantitatively characterize the ambiguity in the CSMA context of the expression “Channel detected idle at an instant t in a MAC entity”. This expression is ambiguous because it only expresses a local view whereas the channel is geographically distributed. For example, a MAC entity i sees the channel idle at the instant t but just before t , another one has just sent a frame and this frame has not arrived at i yet, a collision can occur when i sends its frame. So a local view can be different from a global one which can create collision situations. The concept of “Ambiguity Time Window” represents the maximal duration which is possible between the decision to send a frame by a MAC entity and the inevitability of the occurrence of a collision on this frame. The quantitative characterization depends on the transceiver parameters (τ_{ST} , τ_{TT}) and the maximum propagation time between two nodes (call τ_{PT} this Propagation time).

The “Ambiguity Time Window” is represented on Figure 1. The MAC entity i receives a Transmission Request (TR) from the upper layer at the instant $(t - \tau_{ST})$ to send a data and then makes a CCA during τ_{ST} . We suppose that the channel is detected idle at the instant t and then the MAC entity i decides to send its frame. Its transceiver turns to the Transmit state (during τ_{TT}), and then the frame is transmitted and arrives at the level of the most remote MAC entity j at the instant $(t + \tau_{TT} + \tau_{PT})$.

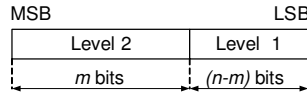
Suppose that the MAC entity j is just, at this instant, finishing to make a CCA (started by a TR at the instant $(t + \tau_{TT} + \tau_{PT} - \tau_{ST})$) and as the channel has been detected idle during τ_{ST} , it decides to send a frame at the instant $(t + \tau_{TT} + \tau_{PT})$. So we have, at this instant, a situation of an inevitable future physical collision. Actually, the beginning of the arrival of the frame coming from the MAC entity i coincides with the instant where the transceiver of the MAC entity j turns around to go in the transmitter state (duration τ_{TT}). During this time τ_{TT} , the MAC entity j is blind. Then at the end of τ_{TT} , the MAC entity j will start to send its frame and we will then have the physical collision.

The duration $(\tau_{TT} + \tau_{PT})$ is the length of the “Ambiguity Time Window”.

2.3 On the Priorities Associated to the Frames

The priorities of the frames are extracted from the values of an identifier (ID) field of n bits. We distinguish two types of priorities:

- Static priorities *i.e.* each flow has a unique priority (specified out of line) and all the frames of this flow have this priority.
- Hybrid priorities with two priority levels (Figure 2); the level 1 is a static priority which expresses the flow priority; the level 2 is a dynamic priority which expresses the frame transmission urgency; the urgency can be the same for all the frames of a flow or depends on the number of the frame in the flow (in these two cases, the urgency is a static priority *i.e.* specified out of line); the urgency can also vary dynamically (for example, if the conditions of application, which uses the flow, change) and then, in this case, the transmission urgency is a dynamic priority.



MSB: Most Significant Bit LSB: Least Significant Bit

Figure 2: Bit field structure (hybrid priority).

2.4 Concept of Tournament

The tournament consists in the comparison of the priorities of the contending frames. This allows to transform a situation which would have been a “collision situation” (if we have the strict CSMA mechanism *i.e.* only based on the scheme “Listen before Send”) into a “winner-loser(s) situation”. There will be only one winner who has the highest priority among the contending frames. The winner can send its frame after the tournament while the losers have to wait until the end of the frame transmission of the winner and restart the tournament.

The good functioning of a tournament is dependent of the BB message durations in the case of the BB message technique, and, of the duration of the ID field in the case CANlike. *It is the consideration of the constraint done by the Ambiguity Time Window which allows to specify correctly these durations.*

We present precisely the tournament for the different protocols in the next sections.

3 CSMA MAC PROTOCOLS BASED ON BB MESSAGES

Call BB-sta and BB-hyb the BB message-based MAC protocols using the static and hybrid priorities respectively.

3.1 BB-sta Protocol

3.1.1 Principle

We have one BB message per frame and its length is proportional to the priority. If we call k the priority of a frame, the BB message of this frame has a length $kTBB$ where TBB is the duration of an energy pulse. All the nodes, which have frames ready to send, send their BB messages. Then we have a tournament among these BB messages. The node which has the longest BB message (*i.e.* the highest priority) is the only winner. When the winner has detected its winner situation, it sends its frame. The losers have to wait until the end of the frame transmission of the winner and then restart a new tournament by sending their BB messages.

3.1.2 Necessity of Channel Observation Times

The start of a tournament must be preceded by an idle channel observation time noted TOBS1. The role of this time is to assure that the channel is globally idle and there is no tournament or transmission in progress. Furthermore, we need another observation time (noted TOBS2) after the sending of BB messages, TOBS2 allows a MAC entity to declare itself winner or loser, the winner detects the channel idle while the losers detect the channel busy.

3.1.3 The Stages of the BB-sta Protocol

The stages represented on Figure 3 are self-explanatory.

3.1.4 Specification of BB-sta Parameters: TBB, TOBS2, TOBS1

TBB duration

Consider the two most remote MAC entities i and j which must send their frames with priorities $(k+1)$ and k respectively. We have to analyze the scenario which expresses the hardest constraint in order to guarantee that, in the hypothesis of a collision situation if we have used the strict CSMA protocol, this situation gives now the situation “MAC entity i is the winner; MAC entity j is the loser”.

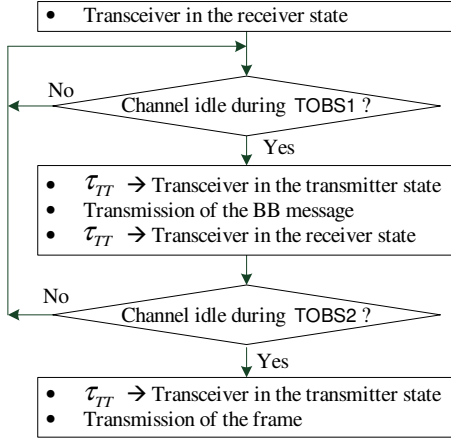


Figure 3: Principle of the BB-sta protocol.

This scenario represented on Figure 4 is one where, by considering the concept of “Ambiguity Time Window”, the MAC entity i decides to send its BB message $(k+1)TBB$ at the time t and the MAC entity j decides to send its BB message $kTBB$ at the time $(t + (\tau_{TT} + \tau_{PT}))$ *i.e.* at the latest time. This expresses the hardest constraint because the arrival of the BB message of the MAC entity j in the MAC entity i has more chances to go beyond the BB message sent by this MAC entity.

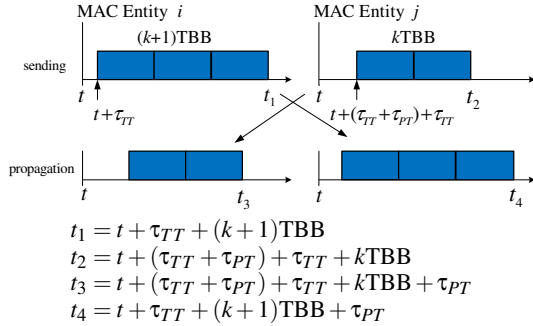


Figure 4: BBsta : Evaluation of TBB.

Then the result “ i : winner, j : loser” in this scenario requires that the two following conditions be satisfied:

- At the MAC entity i : the end of the sending of the BB message generated by i (time t_1) must be later than the end of the propagation of the BB message coming from j (time t_3): $t_1 \geq t_3$. We have $TBB \geq 2\tau_{PT} + \tau_{TT}$.
- At the MAC entity j : the end of the propagation of the BB message coming from i (time t_4) must be later than the end of the sending of the BB message generated by j (time t_2) and the MAC entity j must detect such situation, *i.e.* after the turnaround (τ_{TT}), j must see the chan-

nel busy during at least one τ_{ST} . The condition is $t_4 \geq t_2 + (\tau_{TT} + \tau_{ST})$ which gives $TBB \geq 2\tau_{TT} + \tau_{ST}$.

Then taking into account the two above conditions, we have: $TBB \geq 2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$. We take:

$$TBB = 2\tau_{PT} + 2\tau_{TT} + \tau_{ST} \quad (1)$$

TOBS2 duration

The value of TOBS2 requires, at first, by considering the value of TBB, to evaluate in the loser (MAC entity j) the overtaking of the BB message $kTBB$ sent by j by the BB message $(k+1)TBB$ coming from i . This overtaking is $t_4 - t_2 = 2\tau_{PT} + \tau_{TT} + \tau_{ST}$ which includes one turnaround time (τ_{TT}) and TOBS2. We then get the value of TOBS2:

$$TOBS2 = 2\tau_{PT} + \tau_{ST} \quad (2)$$

TOBS1 duration

Note (Figure 3) that, TOBS2 is surrounded by two turnaround times (one before TOBS2 and the other before the frame transmission by the winner). Then in any tournament, the channel is idle during $(TOBS2 + 2\tau_{TT})$ between the end of the BB message sending and the start of the frame transmission by the winner. Thus it is enough to have TOBS1 higher than $(TOBS2 + 2\tau_{TT})$ at least one τ_{ST} in order to see the channel state correctly. Then we take $TOBS1 = TOBS2 + 2\tau_{TT} + \tau_{ST}$ which gives:

$$TOBS1 = 2(\tau_{TT} + \tau_{PT} + \tau_{ST}) \quad (3)$$

TBB	$2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$
TOBS1	$2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$
TOBS2	$2\tau_{PT} + \tau_{ST}$

Table 1: BB-sta parameters.

Summary

We present in Table 1 the values of the parameters of the BB-sta protocol.

From these parameters, we can deduce the time for accessing the channel (call τ_a) by the winner (suppose that the winner has a priority k). We have:

$$\begin{aligned}
 \tau_a &= TOBS1 + \tau_{TT} + kTBB + \tau_{TT} + TOBS2 + \tau_{TT} \\
 \tau_a &= 4\tau_{PT} + 5\tau_{TT} + 3\tau_{ST} + kTBB
 \end{aligned} \quad (4)$$

3.1.5 An Example of a Time Diagram of BB-sta

We present on Figure 5 a time diagram of BB-sta. We consider four nodes A, B, C, D in which, during the transmission of A, the nodes B, C, D have TRs to send frames with priorities 3, 2, 1 respectively. For the simplicity, we consider $\tau_{PT} = 0$ and B, C, D have already been in the Receiver state before the end of the transmission of A. In these conditions, B, C, D see the channel free at the same time. At the end of the transmission of A, the three nodes B, C, D detect the channel idle during TOBS1, they do the turnaround and then send their BB messages. B has the highest priority so it is the winner and transmits its data after the channel observation TOBS2 while C, D are losers and they have to wait till the end of the frame transmission of B and then they start the channel observation TOBS1 and so on.

3.2 BB-hyb Protocol

3.2.1 Principle

We have two BB messages per frame: a BB_s message and a BB_d one which represent the static and dynamic priority respectively. The contending nodes send first their BB_d messages (that represents the predominance of the urgency), and we can have one or several winners (because nodes can have the same transmission urgency). The losers of this BB_d message tournament will withdraw the tournament while the winners continue the tournament by sending their BB_s messages. The tournament between the BB_s messages provides the global and only winner who has the longest BB_s message. This global winner then sends its frame while all the losers of two sub-tournaments (between BB_d and BB_s messages) will wait until the end of the frame transmission of the winner and then restart a new tournament by sending their BB_d messages.

Considering a frame which has a static priority k_s and a dynamic one k_d . The BB_s and BB_d messages of this frame have the durations $k_s TBB_s$ and $k_d TBB_d$ respectively, in which, TBB_s and TBB_d are the durations of an energy pulse concerning respectively the static and dynamic priority.

3.2.2 Necessary channel observation times

As the same reason in BB-sta, we need an observation time TOBS1 before the start of the tournament in order to avoid intrusions in a tournament in progress. Moreover, we need an observation time (noted TOBS2) after the sending of BB_d messages and another one (noted TOBS3) after the sending of

BB_s messages. TOBS2 and TOBS3 allow a MAC entity to declare itself winner or loser of the corresponding tournament.

3.2.3 Necessity of a Guard Time Associated to the BB_d Messages

Consider a particular tournament between several MAC entities which have a frame to send. Suppose that the frames of these MAC entities have the same dynamic priority k_d which is, furthermore, the highest in this tournament. All the MAC entities relative to this tournament must be winners.

However, having in mind the distance and the asynchronous behavior of the MAC entities, the busy state of the channel at the level of a MAC entity winner can be longer than the BB_d message duration sent by this MAC entity. This results from an overlap between a BB_d message sent and the arrival of a BB_d message from another MAC entity winner.

Then for a MAC entity, it is necessary to add a guard time (noted G) after the sending of a BB_d message in order to, after this guard time, any MAC entity will see the channel idle and declare itself a winner. During G , the MAC entity is blind, *i.e.* no sending, no sensing.

3.2.4 The Stages of the BB-hyb Protocol

The stages presented on Figure 6 are self explanatory.

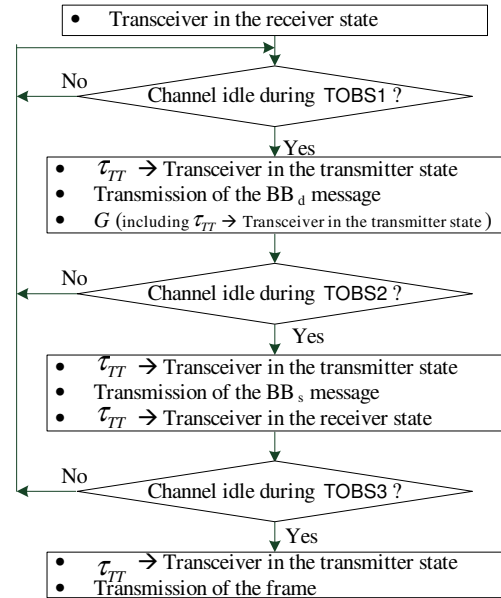


Figure 6: Principle of the BB-hyb protocol.

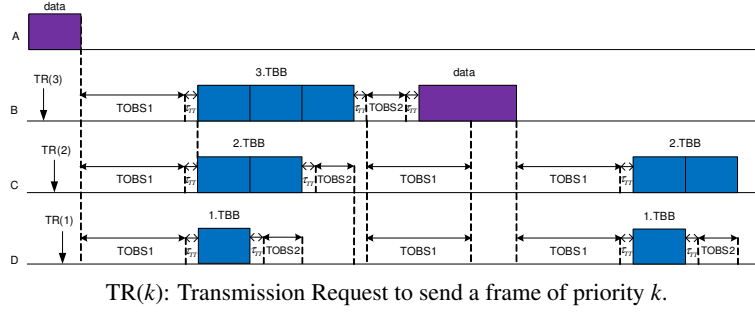


Figure 5: An example of a time diagram of BB-sta.

3.2.5 Specification of BB-hyb Parameters

Guard time G

In order to evaluate the guard time, we have in mind the concept of the Ambiguity Time Window and we consider the following scenario (Figure 7): two MAC entities i and j in which the MAC entity i decides to send the message $k_d TBB_d$ at the time t , whereas the MAC entity j decides to send the message $k_d TBB_d$ at the latest *i.e.* at the time $t + (\tau_{TT} + \tau_{PT})$.

We see that, at the MAC entity i , the end of the propagation of the message $k_d TBB_d$ coming from the MAC entity j (time t_3) is higher than the end of the sending of its message $k_d TBB_d$ (time t_1). The difference is $t_3 - t_1 = 2\tau_{PT} + \tau_{TT}$ which defines the guard time:

$$G = 2\tau_{PT} + \tau_{TT} \quad (5)$$

Note that, as the duration τ_{TT} is included in G , we can consider that the turnaround is done during the guard time and then, after the guard time, the MAC entity can observe the channel. We consider this aspect here.

TBB_d duration

We analyze the same scenario, which has been done in the evaluation of the TBB duration in BB-sta (Sect. 3.1), by considering now the guard time.

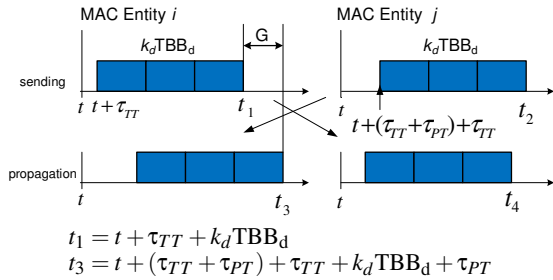


Figure 7: BB-hyb : Evaluation of G .

So the scenario (Figure 8) is: MAC entity i decides to send the BB_d message $(k_d + 1)TBB_d$ at the time t and the MAC entity j decides to send the BB_d message $k_d TBB_d$ at the time $t + (\tau_{TT} + \tau_{PT})$ *i.e.* at the latest time. The result “ i : winner and j : loser” requires the two following conditions:

- At the MAC entity i : the end of the BB_d message sending (time t_1) increased by the guard time (G) must not be overtaken by the end of the propagation of the BB_d message coming from j (time t_3), *i.e.* $t_1 + G \geq t_3$.

$$\Rightarrow TBB_d \geq 0 \quad (6)$$

- At the MAC entity j : the end of the propagation of the BB_d message coming from i (time t_4) must overtake the end of the BB_d message sending generated by j (time t_2), increased by the guard time (G) and the MAC entity j must observe this overtaking, *i.e.* $t_4 \geq t_2 + G + \tau_{ST}$.

$$\Rightarrow TBB_d \geq 2\tau_{PT} + 2\tau_{TT} + \tau_{ST} \quad (7)$$

It is the constraint (7) with the minimum value which dictates the value of TBB_d:

$$TBB_d = 2\tau_{PT} + 2\tau_{TT} + \tau_{ST} \quad (8)$$

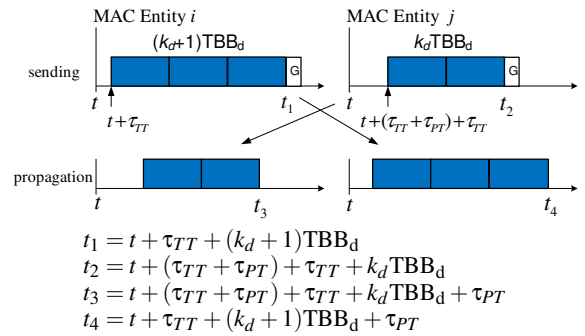
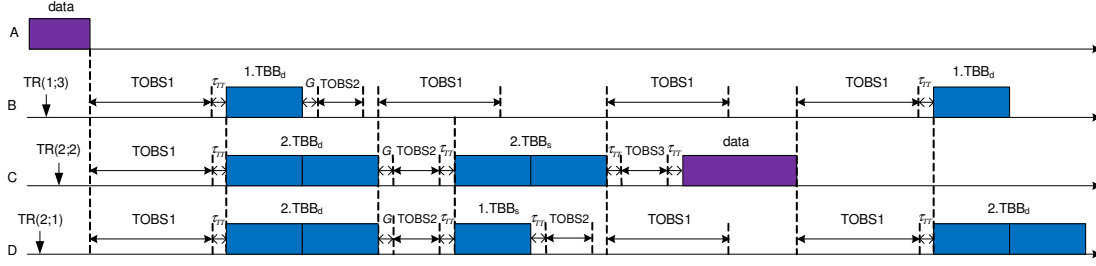


Figure 8: BBhyb : Evaluation of TBB_d.



TR($k_d; k_s$): Transmission Request to send a frame which has a dynamic priority k_d and a static priority k_s .

Figure 9: Example of a time diagram of BB-hyb.

TOBS2 duration

Again we consider the type of analysis for TOBS2 in BB-sta (Sect. 3.1). Considering Figure 8, in the loser (MAC entity j), the overtaking of the BB_d message (sent by this loser) by the BB_d message coming from the winner (MAC entity i) is $(t_4 - t_2)$. This excess includes G and TOBS2, then we have the value of TOBS2:

$$\text{TOBS2} = t_4 - t_2 - G = \tau_{ST} \quad (9)$$

TBB_s and TOBS3 durations

As these values concern the static part of the hybrid priorities, we can take the values obtained for TBB and TOBS2 in BB-sta (Sect. 3.1).

$$\text{TBB}_s = 2\tau_{PT} + 2\tau_{TT} + \tau_{ST} \quad (10)$$

$$\text{TOBS3} = 2\tau_{PT} + \tau_{ST} \quad (11)$$

TOBS1 duration

Note that (Figure 6) TOBS2 is surrounded by G and τ_{TT} , and TOBS3 is surrounded by τ_{TT} and τ_{TT} . Then the channel can be idle during $\max\{G + \text{TOBS2} + \tau_{TT}; \tau_{TT} + \text{TOBS3} + \tau_{TT}\} = 2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$. Thus TOBS1 must be higher than $(2\tau_{PT} + 2\tau_{TT} + \tau_{ST})$ at least one τ_{ST} in order to detect the channel state correctly. We then consider here:

$$\text{TOBS1} = 2\tau_{PT} + 2\tau_{TT} + 2\tau_{ST} \quad (12)$$

Summary

We present in Table 2 the values of the parameters which characterize the BB-hyb protocol. Note that, $\text{TBB}_s = \text{TBB}_d$ (8) and (10), we call TBB these durations.

From these parameters, we can deduce the time for accessing the channel (called τ_a) by the winner (suppose that the winner has a dynamic priority k_d and a static priority k_s): $\tau_a = \text{TOBS1} + \tau_{TT} + k_d \text{TBB} +$

TBB	$2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$
G	$2\tau_{PT} + \tau_{TT}$
TOBS1	$2\tau_{PT} + 2\tau_{TT} + 2\tau_{ST}$
TOBS2	τ_{ST}
TOBS3	$2\tau_{PT} + \tau_{ST}$

Table 2: BB-hyb parameters.

$G + \text{TOBS2} + \tau_{TT} + k_s \text{TBB} + \tau_{TT} + \text{TOBS3} + \tau_{TT}$. We then have:

$$\tau_a = 6\tau_{PT} + 7\tau_{TT} + 4\tau_{ST} + (k_s + k_d) \text{TBB} \quad (13)$$

3.2.6 Remark about the access times (τ_a) of BB-sta and BB-hyb

We can see from (4) and (13) that τ_a strongly depends on the priority value. Hence, if nodes have great priorities, the delays become very important which make the system less efficient. This is the drawback of BB message-based protocols.

3.2.7 An Example of a Time Diagram of BB-hyb

We consider the same example as in the case BB-sta (Figure 5) and the same conditions (*i.e.* B, C, D will start the channel observation TOBS1 at the same time) but now nodes have hybrid priorities ($k_d; k_s$) as represented in Figure 9.

After the channel observation TOBS1, B, C, D find the channel idle, they do a turnaround to be in the Transmit state, and then send their BB_d messages. At the end of the BB_d message transmissions, a guard time G is executed (during G , they do a turnaround to be in the Receive state). After that, they listen to the channel during TOBS2. At the end of TOBS2, B declares itself a loser and withdraws the tournament while C and D declare themselves winners. The next tournament between C and D is done as in the case BB-sta. After the BB_s message tournament, C is the only winner and then sends its data. The losers B and D start a new tournament at the end of the data send-

ing of C (we show only the BB_d message tournament between B and D).

4 CANLIKE PROTOCOL

4.1 Principle

As introduced, CANlike is an adaptation of the MAC protocol of the wired bus CAN to the wireless network.

In the wired bus CAN, MAC entities can send bits and listen to the channel simultaneously. Each MAC entity has a unique ID (identifier) field placed at the beginning of the frame. The ID represents the priority and allows to do the channel access tournament. The tournament is done by a comparison bit by bit of the same rank among the IDs of the frames trying to access the channel. In one bit-by-bit comparison, a bit 0 which is a dominant bit overwrites a bit 1 which is a recessive bit. The MAC entity which has the highest priority will be the only one winner after the tournament and it will send its frame.

In the wireless context, the bus CAN protocol cannot be directly implemented with wireless transceivers since the transceivers cannot transmit and receive simultaneously in the same channel, so we consider the proposal, which has been done in (You et al., 2003), (Pereira et al., 2007): one slot time (duration) is provided for each ID bit, a dominant bit consists in the sending of a carrier pulse during its duration while a recessive bit consists in the sensing/listening of the channel during its duration.

So, in each MAC entity, the tournament on each bit has the following characteristics:

- The MAC entity has a dominant bit: it sends a carrier pulse on the channel and at the end of the sending, it wins by definition the tournament related to this bit and then continues the tournament on the next bit.
- The MAC entity has a recessive bit: either it senses a carrier pulse, then it loses the tournament related to this bit and stops the tournament phase, or it senses nothing (that means that there is no dominant bit sent by another MAC entity) and then it can continue the tournament on the next bit.

It is important to note that the tournament progress is identical whatever the priority type may be (static or hybrid). Then we do not have to consider as with the BB messages, two types of tournament (the one for the static priorities and the one for the hybrid priorities).

4.2 Main Points to Consider

4.2.1 Necessity of the Synchronization Phase

The start of the tournament by a MAC entity (the sending of the first ID bit *i.e.* the MSB bit) must be preceded by the sending of a synchronization (SYN) signal which is an energy pulse (carrier pulse like a dominant bit). The role of the SYN signal is to announce to the other entities the arrival of the ID of this MAC entity and then to provide for them a time reference for the analysis of this ID.

A MAC entity which has detected a SYN signal without itself having sent before a SYN signal do not participate to the tournament.

4.2.2 Necessity of a Guard Time

Having in mind the time interval defined by the Ambiguity Time Window, several MAC entities can send the SYN signals which will be overlapping. Consequently, in each MAC entity among these MAC entities, the end of the SYN signal sending can be overtaken by the end of the SYN signal arrivals. Considering such situation, a MAC entity cannot send the first ID bit immediately after the SYN signal sending. We need to have a guard time following the SYN signal and then we send the first ID bit. The guard time guarantees that after this time, we have a clean (idle) channel *i.e.* there is no more residue of the SYN activity.

We also have the overlap between the ID bits of different rank, then we need to add a guard time at the end of each ID bit.

4.2.3 Necessity of the Channel Observation Time

As in the BB-based MAC protocols, we need an observation time (noted TOBS1) before the start of a tournament.

4.3 The Stages of CANlike

We can now precise the global tournament. All the contending nodes listen to the channel during TOBS1. If the channel is detected idle, all the contending nodes send a SYN signal and then do the tournament by comparing their ID bits from the MSB bit to the LSB bit. The only winner is the one who did not lose on any bit during the tournament. The winner will send its frame while the losers will wait until the end of the frame transmission of the winner and do a new tournament.

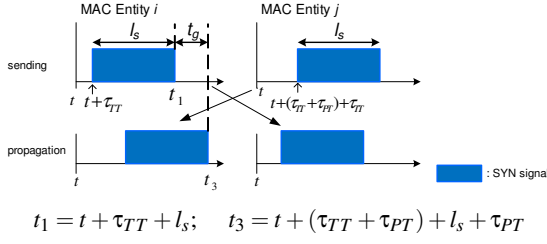


Figure 10: CANlike: SYN signals.

4.4 Specification of the CANlike Parameters

We have to specify the following parameters: length of the SYN signal (noted l_s), the guard time value (noted t_g), the length of the ID bit (noted l_b) and TOBS1.

4.4.1 l_s Duration

The SYN signal must be detected by a receiver MAC entity. Then:

$$l_s \geq \tau_{ST} \quad (14)$$

4.4.2 t_g Value

Suppose that a MAC entity i sends the SYN signal. The duration t_g is the biggest difference between the end of the SYN signal sending and the end of the propagation of a SYN signal of another MAC entity j .

In order to specify the value t_g , we consider the Ambiguity Time Window concept with the most constrained scenario which is represented on Figure 10: the MAC entity i decides to send its SYN signal at the instant t while the MAC entity j decides to send its SYN signal at the instant $t + (\tau_{TT} + \tau_{PT})$ i.e. the latest with respect to the one sent by i .

We can see (Figure 10) that we do not have an overtaking in the MAC entity j because it sends later and that we have an overtaking in the MAC entity i (because it sent earlier). This overtaking ($= t_3 - t_1$) defines the value of t_g . Then we have:

$$t_g = 2\tau_{PT} + \tau_{TT} \quad (15)$$

Remark: As $t_g > \tau_{TT}$, we consider that, during the guard time, a MAC entity makes the turnaround if necessary (that depends on the first ID bit). If the first ID bit is a dominant one, it is not necessary to make a turnaround (because we have still to transmit a carrier pulse and then to stay in the transmitter state; if it is a recessive one, we do the turnaround (because we have to listen to the channel and then to go in the receiver state).

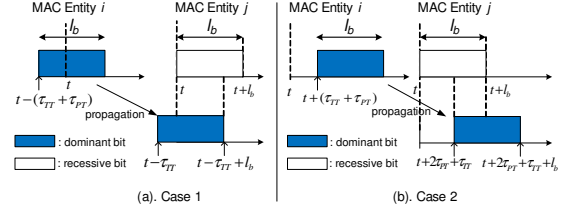


Figure 11: CANlike: Evaluation of l_b .

4.4.3 l_b duration

We analyze the tournament (after the SYN phase) between the first ID bit of the MAC entity i (this bit is a dominant bit) and the first ID bit of the MAC entity j (this bit is a recessive bit) by considering, always in the context of the Ambiguity Time Window, the following scenarios:

- The MAC entity j starts listening to the channel at the time t and end the listening state at the time $t + l_b$.
- The MAC entity i starts the sending of the carrier pulse either (case 1) at the time $t - (\tau_{TT} + \tau_{PT})$ i.e. at the earliest (Figure 11.a), or (case 2) at the time $t + (\tau_{TT} + \tau_{PT})$ i.e. at the latest (Figure 11.b).

For the case 1 (Figure 11.a), the carrier pulse sent by i arrives at j at $t - (\tau_{TT} + \tau_{PT}) + \tau_{PT} = t - \tau_{TT}$ (then we have an arrival of the carrier pulse before the decision to listen to the channel but this event happens during the guard time associated to the SYN signal and then it has no effect) and lasts till $t - \tau_{TT} + l_b$. This arrival must be detected by j (i.e. j sees this arrival during at least one τ_{ST}), the condition is: $t - \tau_{TT} + l_b \geq t + \tau_{ST}$ (i.e. an overlap at the beginning of the listening state) which gives:

$$l_b \geq \tau_{TT} + \tau_{ST} \quad (16)$$

For the case 2 (Figure 11.b), the carrier pulse sent by i arrives at j at $t + (\tau_{TT} + \tau_{PT}) + \tau_{PT} = t + \tau_{TT} + 2\tau_{PT}$ and lasts till $t + \tau_{TT} + 2\tau_{PT} + l_b$. The MAC entity j must detect this carrier pulse arrival (i.e. j sees this arrival during at least one τ_{ST}), the condition is: $t + \tau_{TT} + 2\tau_{PT} + \tau_{ST} \leq t + l_b$ (i.e. an overlap at the end of the listening state) which gives:

$$l_b \geq 2\tau_{PT} + \tau_{TT} + \tau_{ST} \quad (17)$$

Considering the constraints (16) and (17), we need $l_b \geq 2\tau_{PT} + \tau_{TT} + \tau_{ST}$. We take here:

$$l_b = 2\tau_{PT} + \tau_{TT} + \tau_{ST} \quad (18)$$

Note that, in the case 2, the end of the arrival of the carrier pulse is higher than $t + l_b$. The difference is $2\tau_{PT} + \tau_{TT}$ which is equal to t_g (15). We thus need

a t_g added at the end of each ID bit in order to have a clean system (when the tournament between the bits of a given rank starts, there is no thing on the channel from the tournament between the bits of the previous rank).

We also consider that during t_g , a MAC entity does the turnaround if necessary (that depends on the next ID bit value). If the next ID bit and the current one are identical, it is not necessary; if they are different, we have to do the turnaround.

4.4.4 TOBS1 Duration

TOBS1 must be higher than the maximum duration during a tournament where the channel is idle. This extreme case is that we have a channel access of only one MAC entity which has all recessive ID bits. Considering the ID field of n bits, the channel will be idle during $n(l_b + t_g)$, thus $\text{TOBS1} > n(l_b + t_g)$. We take: $\text{TOBS1} = (n + 1)(l_b + t_g)$.

$$\text{TOBS1} = (n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST}) \quad (19)$$

4.5 Summary

We present in Table 3 the values of the parameters which characterize the CANlike protocol. Concerning the duration of the SYN signal l_s , we only indicated its constraint in (14) ($l_s > \tau_{ST}$). We consider that it is not necessary to distinguish its duration from the ID bit duration and then we take the same value ($l_s = l_b$).

$l_b = l_s$	$2\tau_{PT} + \tau_{TT} + \tau_{ST}$
t_g (for l_b and l_s)	$2\tau_{PT} + \tau_{TT}$
TOBS1	$(n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$

Table 3: CANlike parameters.

From these parameters, we can deduce the time for accessing the channel (called τ_a) by the winner. τ_a composes of the observation phase, the SYN phase and the tournament phase:

$$\begin{aligned} \tau_a &= \text{TOBS1} + (l_s + t_g) + n(l_b + t_g) \\ \tau_a &= 2(n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST}) \end{aligned} \quad (20)$$

4.5.1 Remark about the Access Time (τ_a) of CANlike

Contrary to BB-sta and BB-hyb, τ_a of CANlike (20) is constant whatever the priority value may be. This is the interest of this protocol which allows to implement more nodes and applications on the network while having QoS guaranties.

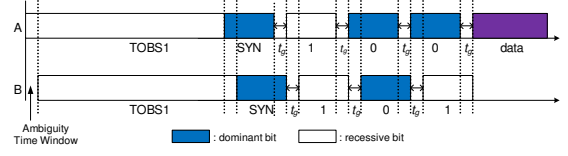


Figure 12: Example of a time diagram of CANlike.

4.6 Example of a Time Diagram of CANlike

We present an example of the tournament of CANlike on Figure 12. We consider an ID field of 3 bits and the tournament of 2 nodes A and B which have the priorities (1; 0; 0) and (1; 0; 1) respectively. Thus A has a higher priority than B. The node B starts the tournament later (of one Ambiguity Time Window) than the node A. At the 1st ID bit, the two nodes, which have recessive bits, find the channel idle; at the 2nd ID bit, the two nodes have dominant bits so they continue the next bit; at the last ID bit, A has a dominant bit so it is the winner by definition while B, which has a recessive bit, finds the channel busy and then B stops the tournament. Then A sends its data.

5 CONCLUSION

In this paper, we have tried to do an exhaustive presentation of the specification process of the collision-free CSMA MAC protocols which are based on the use of priorities which transform collision situations (with a pure CSMA protocol) into winner-loser(s) situations.

We have, in particular, defined the concept of Ambiguity Time Window which formalizes quantitatively (in function of the parameters of the physical layer) the worst case of a collision situation and which then allows to specify the constraint for having winner-loser(s) situations. Thanks to this concept, we have been able to make an understandable and pedagogical presentation of the parameters of the BB message-based protocols and CANlike.

Concerning these two kinds of protocols, as the priorities are generated from a data field (called ID field), it is important to have an idea of the time duration of the exchanges required by the tournament depending on whether we have a protocol based on BB messages or a CANlike protocol. That depends on the number of flows to schedule. If we have a small number (then we need a small number of priorities), a protocol based on BB messages gives exchanges shorter than with CANlike (because here we transmit the complete ID field). However, if we have

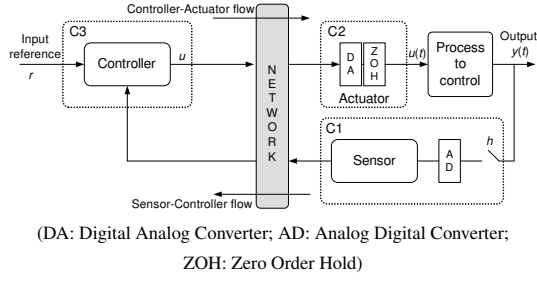


Figure 13: Implementation of a process control application through a network.

a great number of flows (then we need a great number of priorities) and then the BB messages can be very long whereas, with a CANlike protocol, we transmit always the complete ID field (this aspect is still amplified when we consider hybrid priorities). In a general way, a CANlike protocol is a better solution (its implementation, for a given ID field, is independant of the number of flows and of the type of priority: static, hybrid).

We want still to show in this conclusion the interest of the protocol CANlike (with static priorities) considering real-time traffic and in comparison with a protocol used very often in the wireless context (IEEE.802.11-DCF). In this goal, we have, at first, considered an example of a process control application the characteristics of which are: the process to control has the transfer function (Åström and Wittenmark, 1997) $G(s) = \frac{1000}{s(s+1)}$ and the controller is a Proportional Derivative (PD) controller (transfer function: $K(1 + T_d s)$) in order to have a phase margin of 45° which imposes $K = 0.7291$; $T_d = 0.0297$ s. The input reference $r(t)$ is a unity position step and the feedback is made by taking directly the output $y(t)$. The performance of this application *i.e.* the Quality of Control (QoC) is evaluated by means of a cost function ITSE (Integral of Time weighted Square Error) noted J with $J = \int t(r(t) - y(t))^2 dt$. We call J_0 the value which is obtained without the network. Then we have considered the implementation of such four Process control applications (P_i with $i = \{1, 2, 3, 4\}$) through a network where the MAC protocol is either IEEE 802.11-DCF protocol or the CANlike protocol. The analysis of the implementation has been done by using the tool TrueTime (Ohlin et al., 2007) and by considering, for the frame format the frame of IEEE 802.11-DCF (for the CANlike protocol we add also an ID field of 8 bits).

The scheme of the implementation of a process control application through a network is shown on Fig. 13. We have two frame flows: the sensor-controller flow (noted f_{sc}) and the controller-actuator

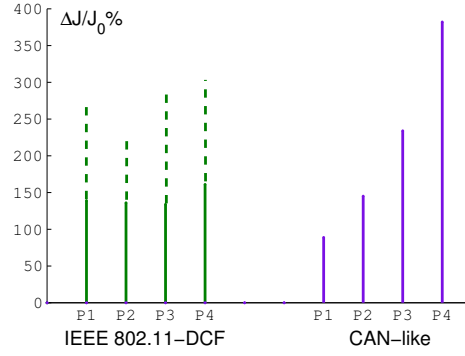


Figure 14: Graphic representation of the QoC ($\Delta J/J_0 \%$).

flow (noted f_{ca}). Then considering the implementation of the four applications through the network, we have eight flows which share the network and compete for its use: f_{sci}, f_{cai} , with $i = \{1, 2, 3, 4\}$. Accounting for the delays which will affect the frames of these flows, the performances obtained for the four applications implemented through a network will be obviously less good than for the application implemented without the network and will also depend on the protocol (IEEE 802.11-DCF, CANlike). Concerning CANlike, we consider the following scheme of static priorities for the frames of the flows f_{sc} and f_{ca} : $\text{prio } f_{ca1} > \text{prio } f_{ca2} > \text{prio } f_{ca3} > \text{prio } f_{ca4} > \text{prio } f_{sc1} > \text{prio } f_{sc2} > \text{prio } f_{sc3} > \text{prio } f_{sc4}$ which means that we consider: importance $P_1 > \text{importance } P_2 > \text{importance } P_3 > \text{importance } P_4$.

The comparison between the performances obtained, by the implementation of the four process control applications, with the two protocols, IEEE 802.11-DCF and CANlike, is made with the performance criteria $\frac{J-J_0}{J_0} \% = \frac{\Delta J}{J_0} \%$ which shows the deviation of the QoC in comparison with the result obtained (J_0) when the implementation is done without network (this comparison shows the influence of the delays induced by the protocols) The higher the value $\frac{\Delta J}{J_0} \%$ is, the more degraded the QoC is.

The Fig 14 shows the results which have been obtained. Concerning the IEEE 802.11-DCF we did 20 simulations and we have represented the mean value by a simple line and the maximum gap between the results of the 20 simulations with a dotted underline. We see that this protocol induces big gaps between the performances, which can be obtained for each application, and then it cannot guarantee a performance (*i.e.* we have random performances). On the other hand, we see that the CANlike protocol guarantees performances which obviously depend on the priority associated to the flows of the process control applications (higher is the priority, better is the perfor-

mance). We have here deterministic performances. Other works have still been done showing the advantage of the hybrid priorities, based on dynamic priorities, for the implementation of a process control application. We do not present here these results for space reasons.

REFERENCES

- 802.11e (2005). 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.
- Åström, K. J. and Wittenmark, B. (1997). *Computer-controlled systems: Theory and design*. Prentice Hall, 3rd edition.
- Crow, B., I., W., L.G., K., and P.T., S. (1997). IEEE 802.11 Wireless Local Area Networks. *IEEE Communications Magazine*, 35:116–126.
- Deng, D.-J. and Chang, R.-S. (1999). A Priority Scheme for IEEE 802.11 DCF Access Method. *IEICE Trans Commun (Inst Electron Inf Commun Eng)*, E82-B(1):96–102.
- Mangold, S., Choi, S., Hiertz, G., Klein, O., and Walke, B. (2003). Analysis of IEEE 802.11e for QoS support in wireless LANs. *IEEE Wireless Communications*, 35:40–50.
- Bosch GmbH (1991). Can specification 2.0, 1991. [online]. Available: www.can-cia.de/fileadmin/cia/specifications/can20a.pdf.
- Ohlin, M., Henriksson, D., and Cervin, A. (2007). *True-Time 1.5 – Reference Manual*. Lund Institute of Technology, Sweden.
- Pal, A., Dogan, A., and Özgüner, F. (2002). MAC Layer Protocols for Real-Time Traffic in Ad-Hoc Wireless Networks. In *Proceedings of the 2002 International Conference on Parallel Processing (ICPP'02)*, pages 539–546, Washington, DC, USA. IEEE Computer Society.
- Pereira, N., Andersson, B., and Tovar, E. (2007). WiDom: A Dominance Protocol for Wireless Medium Access. *IEEE Transactions on Industrial Informatics*, 3(2):120–130.
- Sobrinho, J. and Krishnakumar, A. (1996). Real-time traffic over the IEEE 802.11 Medium Access Control Layer. *Bell Labs Technical Journal*, 10(1):172–187.
- Vaidya, N., Dugar, A., Gupta, S., and Bahl, P. (2005). Distributed Fair Scheduling in a Wireless LAN. *IEEE Transactions on Mobile Computing*, 4(6):616–629.
- You, T., Yeh, C.-H., and Hassanein, H. (2003). A New Class of Collision Prevention MAC Protocols for Wireless Ad Hoc Networks. In *Proc. IEEE International Conference on Communications (ICC'03)*, pages 1135–1140, Anchorage, Alaska, USA.