A Free-Collision MAC Proposal for 802.11 Networks

Omar Alimenti [1,2], Guillermo Friedrich [1], Guillermo Reggiani [1]

[1]SITIC Group – Universidad Tecnológica Nacional – FRBB
Bahía Blanca –Argentina

[2] IIIE - Universidad Nacional del Sur – DIEC
Bahía Blanca –Argentina

iealimen@criba.edu.ar, {gfried,ghreggiani}@frbb.utn.edu.ar

Abstract. *Wireless technologies are a good choice for work in industrial environments, where it is necessary to interconnect mobile systems or it wants to avoid sensors and controllers wiring in plant. However, these technologies present reliability and timing problems inherent in the radio channels, mechanisms for medium access, etc. The standard 802.11e provides two alternatives for medium access (EDCA and HCCA) by differentiating traffic into four Access Categories (ACs). This paper proposes a mechanism for controlling the medium access, so-called WRTMAC, developed from the EDCA scheme of standard 802.11e. The handling of the arbitration inter frame spaces (AIFS) has been modified in order to make deterministic the medium access, even in terms of high traffic next to the saturation of the system.*

1. Introduction

Wireless technologies have become a very attractive option for industrial and factory environments. We can appoint the reduction of time and cost of installation and the maintenance of cabling industrial and their changes. The damage on the wirings and connectors due to the aggressive environments of certain types of industries is another reason. The applications of industrial control that involve some kind of mobile systems, in which data communications must meet requirements of real time and reliability, can benefit from the wireless interconnection [Willig A., Matheus K. and Wolisz A., 2005]. However, it is necessary to consider features of the wireless medium, such as the typical weaknesses of a radio frequency channel (RF), the mobility of some stations, the uncertainty in the time of physical medium access of some protocols, etc.

Despite other types of existing wireless interconnections, we are interested on wireless local area networks (WLAN) based on IEEE 802.11 standard. The Medium Access Control protocol (MAC) is decisive in the performance of the network [Vanhatupa T., 2008]. The 802.11MAC mechanism can operate in two ways: Point Coordination Function (PCF) and Distributed Coordination Function (DCF). PCF, also called free of contention, uses an Access Point (AP) as a network coordinator. In DCF, without centralized control, the nodes compete for the access to the physical medium. In spite of the differences, both modes use the Carrier Sense Multiple Access with Collision-Avoidance (CSMA/CA) mechanism to obtain the access to the medium and transmit. One of the weaknesses of the 802.11 MAC protocol is that it not support differentiated quality of service (QoS) for different types of traffic. For that reason, 802.11e [IEEE Std 802.11e; Part 11, 2005] was developed to support two QoS mechanisms: Enhanced Distributed Coordination Access (EDCA) and Hybrid Coordination Function Controlled Channel (HCCA). The EDCA scheme extends DCF, as it is known in the original standard [IEEE Std 802.11; Part 11, 2007], differentiating four prioritized Access Categories (AC) [Vittorio S. and Lo Bello L., 2007]. In spite of EDCA improves the throughput and the response time with regard to DCF, the reduced amount of AC limits the differentiation of traffic with temporary restrictions [Ferré P., Doufexi A., Nix A. and Bull D., 2004]. This paper proposes changes at the MAC level, based on the standard 802.11e, in order to adequate the EDCA mechanism for real-time industrial applications, generating a number of ACs as devices and/or messages are there in the network [Pereira da Silva M. and Becker Westphall C., 2005], making deterministic the time to access the medium. This mechanism has been called WRTMAC: *Wireless Real- Time Medium Access Control*.

2. DCF and EDCA

A wireless local area network (WLAN) 802.11 type is a broadcast network, characterized by the uncertainty in the medium access time.

DCF is a distributed medium access control scheme, based on the CSMA/CA mechanism. A station must sense the medium before starting a transmission; if the medium remains idle during a random time, the station transmits, otherwise its transmission must be postponed until the end of the current one. DCF distinguishes two techniques: the simplest, the station transmits the frame when it is obtained the access to the medium, and waits the acknowledge (ACK) from the receiver; the other uses an exchange of RTS/CTS frames between sender and receiver, prior to the dispatch of the data, in order to avoid collisions due to the hidden nodes [Bensaou B., Wang Yu and Chi Chung Ko, 2000]. This work is based on the first one.

A collision is difficult to detect in a wireless medium, so a given amount of time named inter-frame space (IFS) is used to control the access to the channel. When sensing indicates that the medium is free, a station must wait a time named distributed inter-frame space (DIFS) after the end of the previous transmission (Figure 1). Then there is a waiting time, named backoff window (BW), whose duration is a random quantity of slots time (ST), between a minimum of 0 and a maximum equal to CW–1. CW is the value of the contention window, which begins with a minimum value CW_{min}, and doubles this value after each collision up to a maximum CW_{max}. When the BW timer reaches zero and if the medium remains free, the station begins its transmission. If the medium becomes busy before BW expires, this timer is frozen until the channel remains idle during a DIFS time. If BW expires in two or more stations at the same time, there will be a collision. After a frame was received satisfactory, the receiver station must wait a time short inter-frame space (SIFS) to send an ACK (Figure 1). If the transmitter station didn't receive the ACK after a SIFS time from the end of its message, interprets that a collision has occurred and will be necessary retransmit. The collisions possibility of this mechanism causes uncertainty about the time needed to realize a transmission.

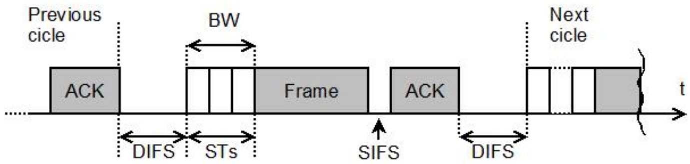

Figure 1. 802.11 DCF Timing

The 802.11e standard introduces the EDCA mode (Figure 2), which proposed a differentiated mechanism of QoS with four ACs: AC_BK (Background) for the lowest priority level (1-2), AC_BE (Best Effort) for the following levels (0-3), AC_VI (Video) for the priorities 4 and 5 and AC_VO (Voice) for the highest (6-7). According to its priority, a frame will be located in one of those four categories. Each AC uses specific values of arbitration IFS (AIFS), CWmín and CWmáx [Willig A., (2008)].

The difference between DCF and EDCA is that, the first does not distinguish types of traffic and, when the medium is free, all stations must wait for the same DIFS before starting its BW timer to contend for the medium access, using all the same CW. However, each type of traffic in EDCA, parameterized for its AC_i, will start its BW timer after sensing the medium idle for a while $AIFS_i$. The AIFS value depends on the AC of the message; therefore an AC of higher priority will have a lower value, having more probability to access the channel. Due to frames with the same AC can coexist in several nodes, collisions can occur and they are resolved in a similar way to DCF

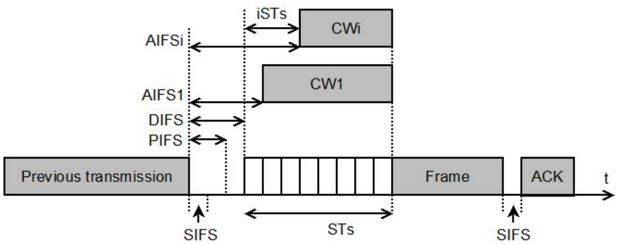

Figure 2. EDCA Timing

The goal of WRTMAC is to develop a collision free MAC method that guarantee the response time, defined as "the time measured from transmission request until the ACK reception". The basic proposal establishes one AC for each type of message, assigning a given $AIFS_i$ to each one. The waiting time prior to a transmission is equal to DIFS plus the $AIFS_i$ according to the type of message *i*.

3. WRTMAC: a Real Time variant to WLAN 802.11

3.1. Basic scheme

The objective that has been established for WRTMAC is a real time deterministic behavior. So, the maximum latency to transmit a frame must be ensured and must be necessary to remove those probabilistic elements own of DCF and EDCA.

EDCA has been the starting point for defining WRTMAC, introducing variants to achieve the target. In that sense, have been established the following patterns of operation:

- Each type of frame has assigned a certain priority, different from any other, in a similar way to the bus CAN [Bosch Robert GmbH, 1991].
- The priority is indicated by a numerical value from zero (maximum priority) and a certain positive number *N* for the minimum. The total amount of priorities should be established by the amount of types of messages that should be handled in the context of a particular application.
- If two or more simultaneous requirements arise, always the frame of the highest priority must be transmitted.
- The logic for controlling the access to the channel has been designed to avoid the occurrence of collisions. However, collisions can occur after intervals of prolonged inactivity, due to the drift between the local clocks of the nodes. The resolution of these collisions should be done in a bounded and predictable time. Also, it has been designed a simple strategy to allow a free-collision operation.

Figure 3 presents the basics of WRTMAC. When a station has a frame to send, it must wait until the medium becomes idle. After a while called "Real-Time Inter-Frame Space" (RIFS), if the medium is still free, the transmission starts. If during the wait, the channel becomes busy, the process will be stopped and should be restarted when the medium becomes free.

In WRTMAC each message has its exclusive RIFS value. Its duration is inversely proportional to the priority it represents. $RIFS_i$ is called the waiting time (backoff) for the priority ***i*** message.

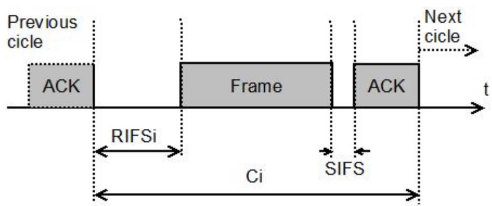

Figure 3. WRTMAC: Basic scheme

WRTMAC determinism is based on that, each message uses a $RIFS_i$ arbitration time, fixed and different from others. This tends to avoid the occurrence of collisions, while ensuring that, in case of contention, the winner will be the higher priority message.

Figure 4 shows the order of three frames, with priorities 2, 3 and 4, contending for the medium access. The three nodes begin the wait, but as $RIFS_2$ is the shortest, $Frame_3$ and $Frame_4$ attempts must be aborted. They are restarted after the end of $Frame_2$ cycle.

$RIFS_i$ duration is calculated based on the values of DIFS and ST. They are established by the selected physical layer (PHY) of the standard, according to (1):

$$RIFS_i = DIFS + i * ST \quad (1)$$

Table 1 shows the values of SIFS, DIFS and ST for different variants of physical layer (PHY):

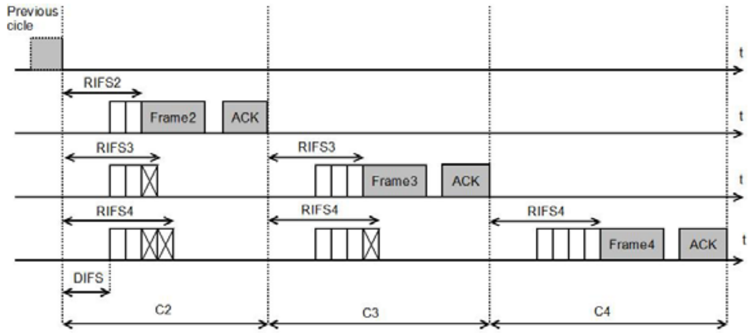

Figure 4. Transmission order according to priority of messages

Table 1. 802.11: PHY variants

PHY	Frec. (GHz)	Rate (Mbps)	SIFS (μs)	DIFS (μs)	ST (μs)
802.11b	2.4	11	10	50	20
802.11g	2.4	54	10	28	9
802.11a	5.8	54	9	16	34

The transmission cycle of priority *i*, composed by $RIFS_i$, the transmission time of the frame *i* (t_{FRAMEi}), SIFS and the transmission time of the ACK frame (T_{ACK}), is called C_i (2):

$$C_i = RIFS_i + t_{FRAMEi} + SIFS + t_{ACK} \qquad (2)$$

The ACK instructs the MAC entity of the transmitter that the frame sent, reached its destination. In general, if ACK is not received a retransmission is not performed, but notifies the upper layers that the transfer has failed (or at least that there is no certainty that has been successful). The decision regarding what actions must be taken is left to the uppers layers; they know the logic and timing constraints of the application. WRTMAC is only responsible for providing deterministic communication service on the maximum latency. Only it would be performed a unique retransmission in case of collision, without affecting the deterministic behavior, as explained in 3.4.

One can see that WRTMAC allows implement a Real-Time scheme of Rate Monotonics (RMS) type [Liu and Layland, 1973], assigning priorities to messages in reverse order of their periods. Knowing t_{FRAMEi} for all messages of a certain real-time system, and assuming that they are periodic, one can set the minimum possible period between transmission requirements (T_i) for a given message m_i, in terms of all other messages m_j of higher priority than m_i (where $j<i$). Adapting the classic formula used to analyze the schedulability of a set of real-time periodic tasks on a processor [Lehoczky J., L. Sha, and Y. Ding, 1989], the minimum period possible for a message of priority *i*, is (3):

$$T_i \geq \sum_{\forall j<i} \left\lceil \frac{T_i}{T_j} \right\rceil C_j + C_i \qquad (3)$$

Where: T_i, T_j, C_i, C_j: Period and transmission cycle of priority messages i and j.

Figure 4 shows messages with periods $T_2 \leq T_3 \leq T_4$. (3) is valid when the network is working with high traffic, i.e. when there is always at least one transfer request pending, awaiting the end of the current transmission. However, depending on the total amount of messages and their periods, there may be long intervals of silence. This requires a special analysis, because the completion of a transfer is the event that reset the counter on each node and allows maintaining synchronism.

Exceeding the time corresponding to RIFS longer without having made any transmission, all nodes must restart their counters. Each node must maintain the timing of activity in the medium, even when there is no requirement, because even a moment prior to the expiration of its RIFS is able to receive a request and transmit it in the current cycle.

3.2. Operation on non-saturation conditions

When the network has extended moments of silence, all nodes must proceed as follows: wait a while corresponding to the duration of the message of the lowest priority plus a ST, and then restart their counters. In fact, as one takes into account the time ST elapsed after $RIFS_N$, the counters $RIFS_i$ are not restarted from zero, but with an initial value ST (equivalent to consider that the timers are reset at the expiry of $RIFS_N$).

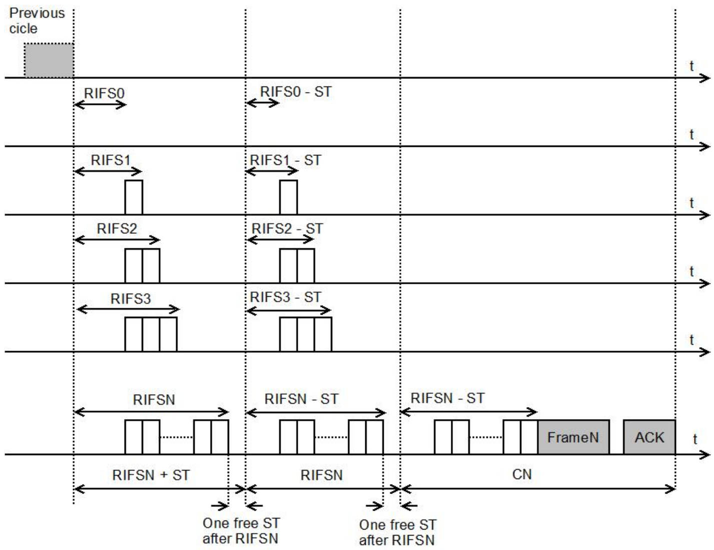

Figure 5. WRTMAC: counters in non-saturation conditions

Figure 5 shows the transmission of a frame after cycles of inactivity. In real operating conditions could succeed longer intervals of silence, maintaining the same concept to restart the counters after each RIFSN period of inactivity in the medium. In case of more extended inactivity, the drift between the locals clocks of each node can lead to a collision condition, as is discussed in 3.4. Another situation to consider is priority inversion, discussed in 3.3. Both, the collision and the priority inversion introduce delays, which must be added to (3) to generalize its expression.

3.3. Priority Inversion

Priority inversion is called the situation that occurs when the transmission of a frame must wait until the completion of a lower priority. Figure 6 shows the almost simultaneous transmission request from $Frame_2$ and $Frame_3$, however, as the requirement of $Frame_2$ released an instant after the end $RIFS_2$, its transmission must wait for the next cycle.

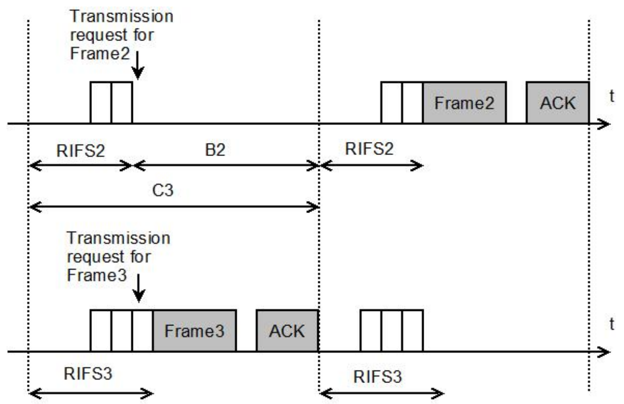

Figure 6. Priority inversion: *$Frame_3$* is transmitted before *$Frame_2$*

As the requirement $Frame_3$ came before the expiry of RIFS3, it is transmitted and completes its cycle C_3. Thus $Frame_2$ was blocked for a while B_2, whose maximum value is $B_2 = C_3 - RIFS_2$. If all frames are considered of lower priority than 2, the maximum block time of $Frame_2$ is:

$$B_2 = máx(C_j) - RIFS_2 \qquad \forall j > 2 \tag{4}$$

In general, for any frame of priority i, time blocking by priority inversion is:

$$B_i = máx(C_j) - RIFS_i \qquad \forall j > i \tag{5}$$

Another deadlock occurs when a frame must wait until the next cycle to be transmitted, because its request arrived a moment after the expiration of its RIFS, having requirements that cause a priority inversion (Figure 7).

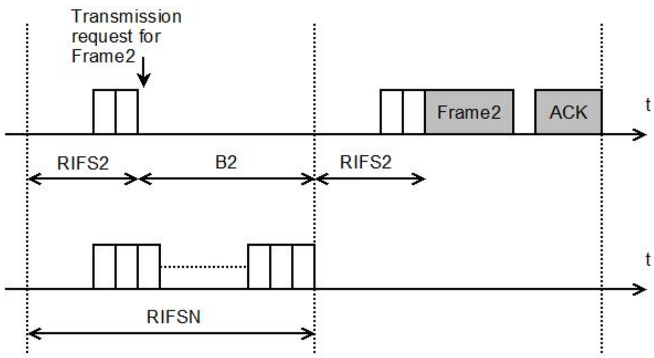

Figure 7. Deadlock of a frame of priority 2 to the end of *$RIFS_N$*

In this case, the deadlock time is $B_2 = RIFS_N - RIFS_2$. As is lower than the priority inversion blocking, remains valid (5).

Based on these considerations, (3) is extended as follows (6):

$$T_i \geq \sum_{\forall j<i} \left\lceil \frac{T_i}{T_j} \right\rceil C_j + C_i + B_i \tag{6}$$

In (6), it isn't taking into account the occurrence of collisions, which is discussed in 3.4.

3.4 Collision by drift of local clocks

When takes place an idle interval of duration greater than or equal to $RIFS_N$, all the nodes must restart their RIFS timers with a periodicity $RIFS_N$.

In case of almost simultaneous requests of consecutive priorities, due to the asynchrony that could exist between clocks of different nodes, it could have cancelled (or reduced almost totally) the difference of one ST between adjacent priority levels, giving rise to a collision (Figure 8).

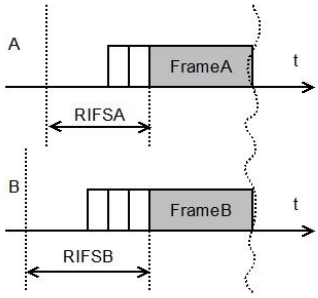

Figure 8. Collision by drift of local clocks

The way in which each node detects a collision depends on the time gap between the ends of the collided transmissions. Nevertheless, after detected a collision and resynchronized the RIFS timers, each node that has been involved in the collision, restarts the process to try a new transmission. This is the unique situation for which a retransmission is allowed, i.e., after a collision following a silence longer than $RIFS_N$.

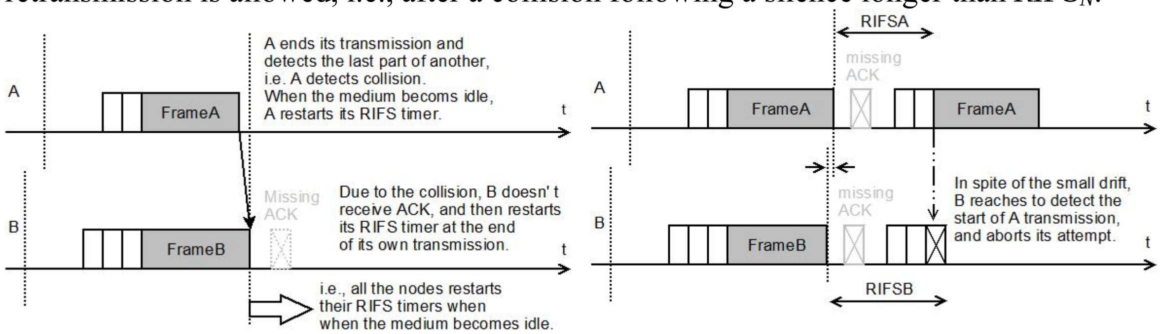

Figure 9. Collisions Type 1 (left) and Type 2 (right)

Figure 9 shows both types of collisions. In Type 1, $Frame_B$ finalizes an amount of time after $Frame_A$, enough to enable node A to sense carrier during the SIFS after its transmission. Node A restarts its RIFS timer with the end of transmission of B. However, B detects the collision by the lack of ACK, and therefore it assumes that the new beginning of its RIFS timer is the end of its own transmission. The remaining nodes that have not been involved with the collision, restart their timers when the medium becomes idle, i. e. after the end of $Frame_B$. In this case, the end of transmission of $Frame_B$ is the event that allows the clock synchronization between all the nodes.

In Type 2, the involved frames finalize with a tiny difference of time, which not allows the detection of the later ending frame by the other node. Then, all the nodes synchronize with the last end of transmission ($Frame_A$ in the example), with the exception of the node that ends in first place its collided transmission ($Frame_B$ in the example). But this gap is smaller than that it is needed to cause a new collision, because it did not allow the sensing of the later end-of-frame by the first finishing node.

After collision detection and clock synchronization, the pending transmissions will be dispatched according to their $RIFS_i$, following the rules of WRTMAC.

3.5 Worst-case delay due to a collision

Figure 10 shows that, after a long idle period a request arrives for the sending of $Frame_i$, but it comes immediately after the expiration of $RIFS_i$. Then, the node must wait until the next cycle to try again. Since there are no pending requests of priorities lower than *i*, all the RIFS the timers will be restarted after the expiration of $RIFS_N$. Until that moment, the delay accumulated by $Frame_i$ is:

$$D_i = RIFS_N - RIFS_i \tag{7}$$

During the next cycle starts the transmission of $Frame_i$, but a collision occurs. Figure 10 shows the worst-case collision delay for $Frame_i$, because is involved the longest frame (the effect would be the same if $Frame_i$ is the longest one).

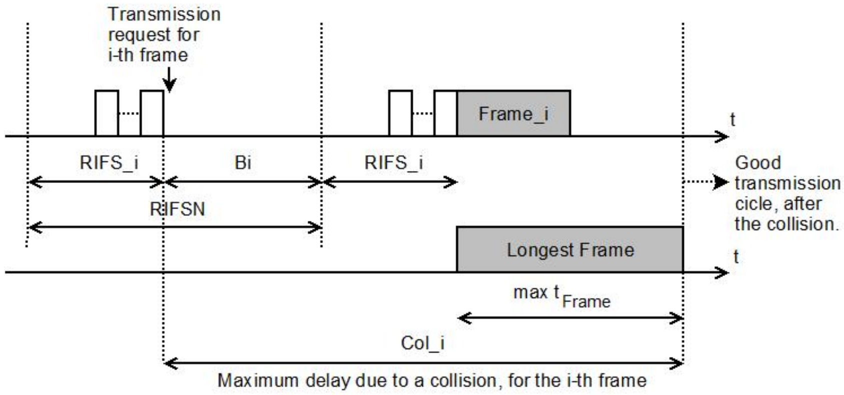

Figure 10. Worst-case delay due to a collision

After the collision, a new cycle starts. If $Frame_i$ continues being the highest priority among those that are awaiting transmission, the delay due to the collision is:

$$Col_i = D_i + RIFS_i + max(t_{FRAME}) \tag{8}$$

Replacing D_i from (7), and as the resulting delay does not depend on frame priority, it is designated generically as *Col*:

$$Col = RIFS_N + max(t_{FRAME}) \tag{9}$$

Now it should be modified the formula (6) to include the delay due to a collision. However, the priority inversion blocking, B_i, and the delay due to a collision, *Col*, are mutually exclusive. It could happen one or the other but not both. Thus, the minimum possible period between requests of the ith-message is:

$$T_i \geq \sum_{\forall j<i} \left\lceil \frac{T_i}{T_j} \right\rceil C_j + C_i + \max(B_i, Col) \tag{10}$$

The formula (10) allows the schedulability analysis for a particular system, based on the amount of messages, their duration and periodicity.

Although the Rate Monotonic scheduling requires that priorities must be assigned in inverted order with respect to the request periods, if some (or eventually all) messages have the same period, it must be assigned different $RIFS_i$ to each one, to allow the medium access arbitration.

3.6. Collision-Free operation

Given that idle intervals greater than $RIFS_N$ can lead to the occurrence of collisions, one way to avoid them could be to avoid the occurrence of such long intervals without transmissions.

A simple strategy could be that, the node designated for the transmission of the lowest priority frames, always should perform a transmission. Therefore, if at $RIFS_N$ timer expiration it does not have a pending request for the transmission of a message, it must send a dummy frame, whose sole purpose is to occupy the medium, allowing the synchronization of all nodes with the end of this transmission.

Using this simple strategy, becomes valid the formula (6) to establish the minimum possible period for a given message *i*.

4. Performance evaluation

Given the initial motivation that led to the development of WRTMAC, its application in industrial control systems, usually based on small periodical messages, they was evaluated the maximum number of messages that could be driven by a network of this type and / or the shortest feasible period for each one. Also, the utilization factor was calculated, defined it *as the fraction of the total time that the medium is used to transport data*.

It has been considered that the network is used for the transmission of a given set of messages of equal size and period. It has been selected the 802.11b physical layer at 11 Mbps, with long preamble (192 microseconds); message payload of 50, 100 and 1500 bytes (plus 36 bytes of header), and 14 bytes of ACK. Also, it has been evaluated two options: one of this based on Formula (10) –with collisions– and the other based on Formula (6) –collision-free–. The utilization factor has been calculated for the free-collision mode. The results are showed in Table 2.

It is observable that there are no significant difference between payloads of 50 and 100 bytes (typical sizes for supervisory and control systems).

Also, there is a small improvement by applying the free-collision model. Hence, it could be simplified the MAC mechanism, eliminating the handling of collisions and its associated retransmissions.

Moreover, it can be seen that for long messages (1500 bytes) the minimum period does not increase proportionately (eg. 77 ms for 64 messages of 100 bytes each; 143 ms for 64 messages of 1500 bytes each). It is allowed to estimate that this network could be used with a mixing of short messages (monitoring and control) and long messages (data, images, etc.), with a small impact on the real-time performance.

Table 2. Minimum period and utilization factor, based on quantity and size of messages

		Minimum Period (ms)		Utilization Factor (%)
Payload (bytes)	N° of messages	With collisions	Collision-free	
50	8	6	4	4.8 %
	16	12	9	4.8 %
	32	28	27	4.3 %
	64	75	74	3.1 %
	128	232	230	2.0 %
100	8	6	6	9.7 %
	16	12	12	9.7 %
	32	29	29	8.0 %
	64	78	77	6.0 %
	128	237	234	3.9 %
1500	8	15	15	58.1 %
	16	30	30	58.1 %
	32	63	62	56.3 %
	64	144	143	48.8 %
	128	368	366	38.1 %

As can be expected, the utilization factor increases with the payload size, because the influence of the overhead due to headers, ACK, etc. is reduced. However, the utilization factor decreases significantly as the total number of messages increases. This is due to the growing overhead associated with the different RIFSs needed to arbitrate the medium access. Based on this situation, a way to improve the network performance could be to group several messages, sharing the same RIFS. However, as RIFS is used to arbitrate the medium access, the RIFS sharing could be possible only when there are more messages than nodes. In these cases, as it is not possible a collision between messages originated in the same node, it could be possible that several messages share the same RIFS value. This will be analyzed in a future work.

5. Conclusions

WRTMAC (Wireless Real Time Medium Access Control) is a proposal that implements a MAC based on EDCA concepts of the standard 802.11e, in order to achieve a mechanism for distributed wireless medium access, allowing predictable access time to the medium of the devices in the network. Changes are proposed to make suitable the EDCA mechanism, in order to generate as Categories Access (AC) as devices and/or messages are in the network. This will be deterministic the access time to the medium. This proposal could bring the EDCA mechanism for real-time industrial applications.

It was shown that WRTMAC allows the implementation of a Rate Monotonic Scheduling (RMS) scheme, since it is possible to know the minimum feasible period of a message, according to all messages.

The proposal also shows that, despite collisions can occur, the collided frames will be retransmitted, and the collision will be solved in a bounded time. Moreover, WRTMAC can operate in a free-collision mode.

Furthermore, when evaluating the performance of WRTMAC over traffic patterns typical of industrial application networking, it was noted that this mechanism provides an adequate performance without times uncertainties. In addition it was shown that the network could be used by combining short and long messages (for monitoring, control and general data, images, etc.), without greatly real-time performance degradation.

As it was analyzed, the increase in the number of RIFS diminishes the utilization factor of the network, due to the RIFS overhead. Hence, future works will be oriented to share the same RIFS between several messages originated in the same node. The goal will be to develop a methodology to establish the minimum number of RIFS needed for a given set of messages.

References

Willig A., Matheus K. and Wolisz A. (2005), “Wireless Technology in Industrial Networks”, Proceedings of the IEEE, Vol. 93, No. 6 (June), pp. 1130-1151.

Vanhatupa T. (2008), “Design of a Performance Management Model for Wireless Local Area Networks”, PhD Thesis.

“IEEE Std 802.11e; Part 11 (2005): Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications and Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements”.

“IEEE Std 802.11; Part 11 (2007): Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”.

Vittorio S. and Lo Bello L. (2007), “An Approach to Enhance the QoS Support to Real-Time Traffic on IEEE 802.11e Networks”, 6th Intl Workshop On Real Time (RTN07) Pisa Italy.

Ferré P., Doufexi A., Nix A. and Bull D. (2004), “Throughput Analysis of IEEE 802.11 and IEEE 802.11e MAC”, WCNC, IEEE Communications Society.

Pereira da Silva M. and Becker Westphall C. (2005), “Performance Analysis and Service Differentiation in the MAC SubLayer of IEEE 802.11e Ad Hoc Networks”, Proceedings of the Advanced Industrial Conference on Telecommunications, IEEE.

Bensaou B., Wang Yu and Chi Chung Ko (2000), “Fair Medium Access in 802.11 based Wireless Ad-Hoc Networks”, IEEE/ACM The first Annual Workshop on Mobil Ad hoc Networking e Computing (MobiHoc´00), Boston, EUA.

Willig A.(2008), “Recent and Emerging Topics in Wireless Industrial Communications: A Selection”, IEEE Transactions On Industrial Informatics, V4 Nº 2.

Robert Bosch GmbH (1991), “CAN Specification 2.0”. *www.semiconductors.bosch.de*

Liu and Layland (1973), “Scheduling algorithms for multiprogramming in a hard real-time environment”, Journal of the ACM, Vol.20 Nº 1, pp. 46-61.

Lehoczky J., L. Sha, and Y. Ding (1989), “The rate monotonic scheduling algorithm: Exact characterization and average case behavior”. Proc. IEEE Real-Time Systems Symposium, pp. 166-171.