

## PROBABILISTIC TIMING ANALYSIS OF THE h-BEB COLLISION RESOLUTION ALGORITHM<sup>1</sup>

**Ricardo Moraes, Francisco Vasques**

*Department of Mechanical Engineering, University of Porto  
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal,  
e-mail: {rmoraes, vasques}@fe.up.pt*

*Abstract:* Ethernet networks are becoming increasingly popular in industrial computer-controlled systems, as they allow for a single network protocol at both the higher and the lower levels of an industrial communication infrastructure. Despite the introduction in the early 90s of a full-duplex operating mode, numerous industrial Ethernet networks still operate in heterogeneous environments, with Ethernet Switching Hubs interconnecting both independent node stations and industrial Ethernet Repeater Hubs. Among node stations interconnected by a Repeater Hub, the network still operates in the traditional shared Ethernet mode; that is, collisions are solved by means of a probabilistic contention resolution algorithm *i.e.*, the medium access is inherently non-deterministic.

In this paper, it is analyzed an enhanced collision resolution algorithm for shared Ethernet networks, referred as high priority Binary Exponential Backoff (h-BEB). Such algorithm allows the coexistence of Ethernet standard devices together with modified (real-time) devices in the same network segment. Both the analytical and the simulation timing analysis show that the h-BEB algorithm guarantees a maximum access delay that is significantly smaller than for the case of standard Ethernet stations. Such enhanced collision resolution algorithm enables the traffic separation between standard and modified (real-time) stations, and is therefore able to guarantee a real-time communication behavior in unconstrained traffic environments.

Keywords: Ethernet communication; Real-time communication.

### 1. INTRODUCTION

Multiple fieldbus network technologies have been proposed and developed to interconnect sensor and actuators to controllers in the industrial environment, as a consequence of the need for specific communication networks in the plant floor. In spite of the adequacy of some of these fieldbus technologies for many type of applications, the use of different/ multiple technologies has obvious disadvantages: high cost, difficult integration and even the incompatibility between standard devices from different producers [1]. At the upper industrial communication levels (office domain), Ethernet has established itself as the most used communication technology, resulting in low component prices caused by the mass production of these components [2].

When Ethernet networks started to be used also in the plant floor, higher speed and low cost for the communication controllers were the major

motivation. However, traditional shared Ethernet systems, with its simple CSMA/CD medium access protocol, do not allowed real-time capability. Consequently, diverse commercial companies have developed extensions to the legacy Ethernet standards and now multiple systems have the potential to fulfill real-time Ethernet specifications. A brief analysis of the state-of-the-art in Industrial Ethernet solutions is given in Section 6.

Despite the introduction in the early 90s of a full-duplex operating mode for Ethernet networks, numerous industrial Ethernet networks still operate in heterogeneous environments, with Ethernet Switching Hubs interconnecting both independent node stations and industrial Ethernet Repeater Hubs. Consequently, among node stations interconnected by Repeater Hubs, the network still operates in the traditional shared Ethernet mode; that is, collisions are solved by means of a probabilistic contention resolution algorithm. This means that heterogeneous

---

<sup>1</sup> This work has been partially supported by IDMEC and by FCT (project ADVANSYS and BD 13203/2003).

networks are not able to provide a real-time communication service.

### 1.1. Rationale for the h-BEB algorithm

Multiple techniques have been developed to provide real-time communication services in shared Ethernet networks. Such techniques are typically based on either: *avoiding collisions*, by controlling the medium access rights of each station (TDMA scheme, token passing, etc), or *ensuring a deterministic collision resolution*, by modifying the collision resolution algorithm. A third approach (that is not deterministic) is to reduce the number of occurring collisions, enhancing the network responsiveness to real-time message requests.

The drawback of such traditional approaches is that they rule out the coexistence of Ethernet standard stations together with modified (real-time) stations in the same network segment. This means that legacy shared Ethernet systems would not be able to support real-time communications without extensive modifications.

To address this problem, it has been proposed in a previous paper [3] the use of a modified collision resolution algorithm, referred as the “high priority Binary Exponential Backoff (h-BEB)” algorithm. This algorithm allows Ethernet standard stations to coexist with at most one modified (real-time) station in the same network segment, imposing a higher priority to the privileged traffic. This mechanism has been extended in a subsequent paper[4], where it has been proposed the use of a virtual token passing procedure, allowing multiple h-BEB (real-time) stations to coexist with multiple standard Ethernet stations in the same network segment, and still imposing a higher priority for the transfer of privileged traffic.

### 1.2. Paper structure

In this paper, we address the timing analysis of the h-BEB collision resolution algorithm. Section 2 reviews the BEB collision resolution algorithm used in standard Ethernet and describes the h-BEB algorithm. Section 3 addresses the timing analysis of this new algorithm. In Section 4, it is summarized the exact performance analysis in heavily loaded network scenarios. Afterwards, the comparative analysis is done in Section 5; it considers a shared Ethernet environment, where multiple stations are interconnected with a special station; the latter implementing either the h-BEB algorithm (enhanced Ethernet mode) or the BEB algorithm (traditional Ethernet mode). the maximum access delay time is then evaluated, demonstrating that the h-BEB collision resolution algorithm is adequate to support soft real-time applications. Section 6 presents a brief overview of real-time industrial Ethernet solutions. Finally, the paper is concluded in Section 7.

## 2. THE HIGH PRIORITY BINARY EXPONENTIAL BACKOFF ALGORITHM

The CSMA/CD (Carrier Sense Multiple Access with Collision Detection) protocol is the protocol implemented at the MAC layer of both ANSI/IEEE 802.3 [5] and Ethernet local area networks. For a 10/100 Mbps Ethernet implementation, the following set of parameters is used:

Table 1: Ethernet parameters.

	Values	
<b>SlotTime</b>	512 bit times	64 byte times
<b>InterFrameGap</b>	96 bit times	12 byte times
<b>AttemptLimit</b>	16	-
<b>BackoffLimit</b>	10	-
<b>JamSize</b>	32 bit times	4 byte times
<b>MaxFrameSize</b>	12144 bits	1518 bytes
<b>MinFrameSize</b>	512 bits	64 bytes
<b>AddressSize</b>	48 bits	6 bytes

Basically, the CSMA/CD protocol works as follows (Figure 1a): when a station wants to transmit, it listens to the transmission medium. If the transmission medium is busy, the station waits until it goes idle; otherwise, it transmits immediately. If two or more stations simultaneously begin to transmit, the transmitted frames will collide. Upon the collision detection, all the transmitting stations will terminate their own transmission and send a jamming sequence<sup>2</sup>. When the transmission is aborted due to a collision, it will be repeatedly retried after a randomly evaluated delay (backoff time), until it is either successfully transmitted, or definitely aborted (after a maximum number of 16 attempts) [5].

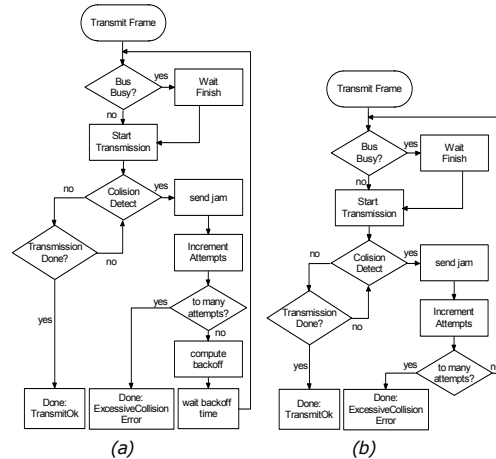


Figure 1. CSMA-CD protocol with BEB resp. h-BEB collision resolution algorithms.

The backoff delay is evaluated by locally executing the Binary Exponential Backoff (BEB) algorithm, which operates as follows: after the end of the jamming sequence, the time is divided into discrete slots, whose length is equal to the slot time. The backoff time is given by  $t_{backoff} = r \times T$ , where  $r$  is a

<sup>2</sup> More accurately, when detecting a collision, the station always finishes the transmission of the Preamble and the Start of Frame Delimiter (64 bits), if these have still not been completely transmitted. Afterwards, it transmits a jamming sequence (32 bits), and then stops.

random integer in the range  $0 \leq r \leq 2^k - 1$ ,  $k$  is the smaller of  $n$  or 10 ( $n$  is the number of retransmission attempts) and  $T$  is the slot time in seconds. This means that the station will wait between 0 and  $2n-1$  slot times. After 10 attempts, the waiting interval is fixed at 1023 slot times, and finally after 16 attempts the transmission is discarded.

On the other hand, a station implementing the h-BEB algorithm operates as follows (Figure 1b): whenever there is a collision, the station immediately starts to transmit (backoff interval equal to 0). This behavior guarantees the highest transmitting probability to the h-BEB station, as it will always try to transmit its frame in the first slot, while all the other stations will wait between 0 and  $2n-1$  slot times.

The h-BEB collision resolution algorithm can be used to support real-time traffic separation, as the traffic generated by the h-BEB station will be always transferred prior to the traffic generated by the other stations. This behavior is highly adequate to, for instance, real-time video/voice transferring applications in legacy shared Ethernet networks. By simply plugging a notebook computer with the modified hardware to the network, it becomes possible to transfer traffic at a higher priority than the traffic generated by all the other stations.

### 3. TIMING ANALYSIS

In this section, the timing analysis of the h-BEB collision resolution algorithm is presented, for the case of a 10 Mbps shared Ethernet scenario. Such analysis can easily encompass a 100 Mbps scenario, using the timing parameters presented in Table 1.

First of all, there is the need to analyze the response time of a shared Ethernet network; that is, the time interval that it takes to transfer a message in a shared Ethernet network.

Consider a two-collision scenario (Figure 2). At instant  $t_0$  station A has a message ready to be transferred ( $P_A$ ), but at instant  $t_0 - \varepsilon$ , another station starts to transmit a 1518-byte message ( $P_N$ ), which is the longest Ethernet message. Station A will wait for the completion of both the message  $P_N$  and the Inter Frame Gap ( $I_1$ : 12 byte times), before attempting to transmit again (that is, 1530 byte times). If a collision occurs during the transfer of the first 64 bytes of message  $P_A$ , a jamming sequence will be broadcasted ( $J_1$ : 4 byte times) and, according to the BEB algorithm, the stations involved in the collision will select a random backoff time (0 or 1 slot time).

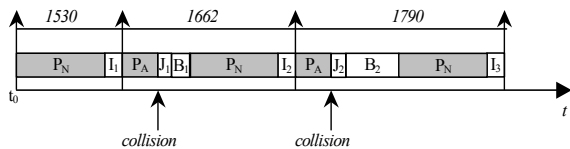


Figure 2: Worst-case 2-collision scenario solved by the BEB collision resolution algorithm.

Considering that station A selects a backoff delay of 1 slot time (64 byte times) and the other station wins

the medium access, a new  $P_N$  message (1518 bytes) may be transferred. Therefore, station A will need to wait again for the completion of both the new  $P_N$  message and the Inter Frame Gap ( $I_2$ : 12 byte times); that is, it must wait  $(64+4+64+1518+12)=1662$  byte times before attempting to transmit for the second time. If a second collision occurs, a jamming sequence will be broadcasted again and, station A may now need to wait during a backoff time of 3 slot times (192 byte times). Therefore, it may need to wait  $(64+4+3 \times 64+1518+12)=1790$  byte times before attempting to transmit for the third time, if a longest  $P_N$  message wins the second collision resolution round. The cumulative result (from  $t_0$  up to the beginning of the third attempt) is then of 4982 bytes or 3,9856 ms (squared box result in Table 2).

On the other hand, a station implementing the h-BEB collision resolution algorithm is characterized by always trying to transmit its frame in the first slot (Figure 3). The worst-case scenario is when, at instant  $t_0$ , station A has a message ready to be transmitted ( $P_A$ ), but at instant  $t_0 - \varepsilon$ , another station starts to transmit a 1518-byte message ( $P_N$ ). In such case, the station will wait for the completion of both the message  $P_N$  and the Inter Frame Gap ( $I_1$ : 12 byte times), before attempting to transmit again (that is, 1530 byte times). If during the transfer of the first 64 bytes of message  $P_A$  a collision occurs, a jamming sequence will be broadcasted ( $J_1$ : 4 byte times) and, station A will need to wait again during an Inter Frame Gap ( $I_2$ : 12 byte times).

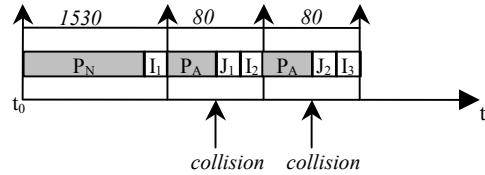


Figure 3: Worst-case 2-collision scenario solved by the h-BEB collision resolution algorithm.

Afterwards, according to the h-BEB algorithm, station A will start to transmit its message. If a second collision occurs, a new jamming sequence will be broadcasted and station A will wait during an Inter Frame Gap, before starting to transmit again. The cumulative result (from  $t_0$  up to the beginning of the third attempt) is then 1690 bytes or 1,3520 ms (rounded box result in Table 2).

Table 2 illustrates the maximum delay to start transferring a message frame after  $i$  consecutive collisions when using, respectively, the BEB and h-BEB collision resolution algorithm.

Figure 4 illustrates the results from Table 2, in a semi logarithmic scale. For the h-BEB case, the maximum delay to start transferring a frame is significantly smaller than for the BEB case. More significantly, such maximum delay is almost constant, which is particularly adequate for the transfer of real-time messages in shared Ethernet environments.

Table 2: Maximum delay to start transferring a message frame – BEB vs. h-BEB.

Retry Number	Max delay (# slots)		Max cumulative delay (# slots)		Max delay (ms)	
	BEB	h-BEB	BEB	h-BEB	BEB	h-BEB
1	1	1	1	1	2,5536	1,2880
2	3	1	4	2	3,9856	1,3520
3	7	1	11	3	5,6224	1,4160
...						
6	63	1	120	6	15,038	1,6080
...						
10	1023	1	2036	10	118,2512	1,8640
...						
14	1023	1	6128	14	332,8752	2,1200
15	1023	1	7151	15	386,5312	2,1840
16	discard frame					

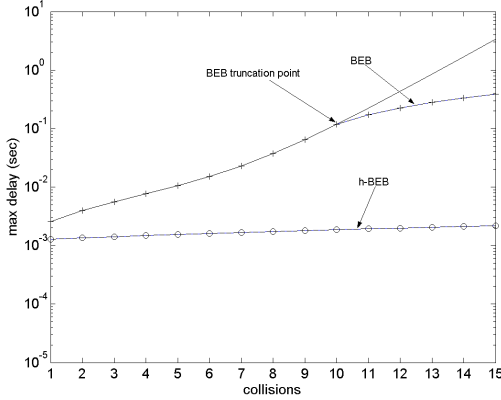


Figure 4: Maximum access delay - BEB vs. h-BEB.

However, as there is still the possibility of a message frame being discarded after 16 failed transmission attempts, there is the need to investigate the probability of such occurrence. Such probability is evaluated in Section 5, both analytically (for an heavily loaded network scenario) and by simulation.

#### 4. EXACT PERFORMANCE ANALYSIS IN HEAVILY LOADED NETWORK SCENARIOS

One of the first Ethernet performance analysis was presented by Metcalfe and Boggs [6], where the authors presented an exact probabilistic analysis for heavily loaded network scenarios. In that analysis, a constant retransmission probability for each slot has been assumed, and the successful retransmission probability (on the next slot) has been considered to be equal to a constant:  $p$ . Such probability  $A$  is maximized when  $p=1/K$  (equal probability of successful retransmission). Such assumption is an interesting approximation for the real backoff function, as has been shown in multiple simulation studies (e.g. [7] [8]). Thus,

$$A = (1 - 1/K)^{K-1} \quad (1)$$

The probability that the contention interval will be exactly  $n$  slots is:

$$P_n = A \times (1 - A)^{n-1} \quad n \geq 1 \quad (2)$$

Obviously, the assumption that each station transmits with an equal probability is not suitable for the

analysis of the h-BEB algorithm, as in the h-BEB case one of the stations (the privileged station) transmits at a higher probability.

Therefore, new and adequate formulae have been devised to perform the probabilistic analysis of the h-BEB collision resolution algorithm. In [3], it has been demonstrated that the probability of the h-BEB station sending a message up to the  $j^{\text{th}}$  collision round (after an initial collision), is given by:

$$P(n, N) = \sum_{j=0}^N (-1)^j \frac{N!}{j!(N-j)!} \times 2^{-jn} \quad (3)$$

where  $n$  is the number of collision resolution rounds, and  $N$  is the number of BEB stations in the network ( $N+1$  is the total number of stations).

#### 5. COMPARATIVE ANALYSIS

A comparative timing analysis of BEB vs. h-BEB algorithms has been performed. For the case of a heavily loaded network scenario (pessimistic case), analytical results enable the evaluation of the maximum access delay vs. the transmission probability. For more realistic load scenarios (intermediate load cases), a more comprehensive analysis of the access delay is done by simulation.

Two cases are analyzed: a *small population* scenario that considers a network with 5 stations, where 4 standard Ethernet stations are interconnected with a *special station* implementing either the h-BEB (enhanced Ethernet mode) or the BEB (traditional Ethernet mode) collision resolution algorithms; a *large population* scenario extends the small population case to 65 interconnected stations.

##### 5.1 Exact Timing Analysis for the Heavily Loaded Network Scenario

Considering the case of a heavily loaded network scenario, the transmission probability of the special station may be obtained from equations (2) and (3), for, respectively, the traditional and the enhanced Ethernet modes. Such transmission probability,  $P_n$  or  $P(n, N)$ , depends on the number of collision resolution rounds  $n$ . When combining the transmission probability for a given collision resolution round, with the maximum access delay (Table 2) for such collision resolution round, it becomes possible to associate a probability of occurrence to each number of maximum access delay.

Such maximum access delay vs. transmission probability is illustrated in Figures 5 and 6, for both the small and large population scenarios, comparing the traditional and the enhanced Ethernet modes.

Consider the enhanced Ethernet mode. It can be seen that, after a small number of collision resolution rounds, the transmission probability is larger than 95% for both the small and large population scenarios. From Equation (3), the transmission probability is larger than 95% when more than 6 or 10 collision resolution rounds are considered,

respectively for the small and large population scenarios:  $P(7,4) = 0,969$  and  $P(11,64) = 0,967$ . As a consequence, the maximum access delay to start transferring a message frame in 95% of the cases is 1,61ms and 1,86ms, respectively for the small and large population scenarios (Table 2). Thus, it is evident that the h-BEB algorithm is clearly adequate to support soft real-time applications.

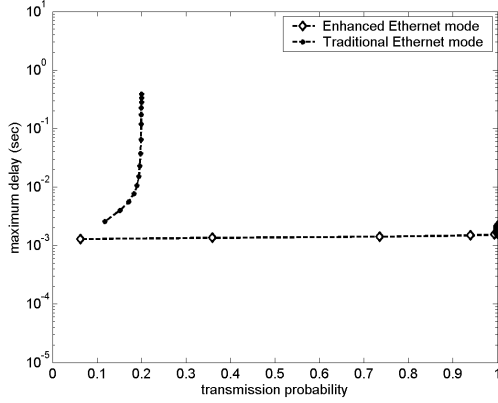


Figure 5: Maximum access delay vs. Transmission probability (small population scenario).

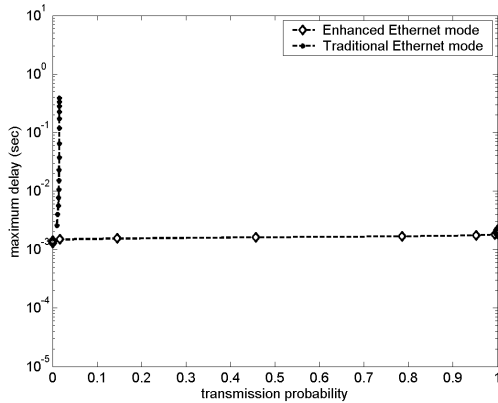


Figure 6: Maximum access delay vs. Transmission probability (large population scenario).

On the other hand, it is of utmost importance to focus on the probability of a message frame being discarded by the h-BEB algorithm, whenever the number of collision resolution rounds exceeds 15. Such probability can be easily evaluated by means of Equation (3), as the probability of a message being discarded is equal to the probability of the h-BEB station not being able to send the message up to the 15<sup>th</sup> collision round, *i.e.*, it is equal to  $1-P(15,N)$ . Such probability is equal to  $1,22 \times 10^{-4}$  and  $1,95 \times 10^{-3}$ , respectively for the small and large population scenarios. Such results are consistent with the claim that the h-BEB algorithm is able to support most part of the soft real-time applications, as they confirm a rather small probability of any message being discarded.

The other set of results is related to the traditional Ethernet mode. For such case, the maximum access delay is significantly higher, even for reduced transmission probabilities. Moreover, the transmission probability is bounded to rather small

numbers (0,20 and 0,015, respectively for the small and large population scenarios), as a constant retransmission probability for each slot has been assumed [6]. This means that the probability of a message frame being discarded (0,80 and 0,985) when using the BEB collision resolution algorithm impairs the support of almost any kind of application in heavily loaded network scenarios.

Nevertheless, it must be considered that such kind of exact timing analysis addresses a rarely occurring case, as it is based on the assumption that, at the start of any transmission attempt, all the network stations participate in the contention process (heavily loaded network scenario). For more realistic load scenarios (intermediate load cases), the performance analysis must be done by simulation, which enables a more comprehensive analysis of both the BEB and the h-BEB algorithms.

### 5.2 Timing Analysis by Simulation

A simulation model was implemented using the *Network Simulator (NS-2)* tool [9], which is a shareware discrete event simulator specially suited for the network performance analysis. For the BEB collision resolution algorithm, a station process implements directly the IEEE 802.3 standard, which is already available in the NS-2 tool. For the h-BEB collision resolution algorithm, a station process has been built according to the h-BEB specification described in Section 2. The implemented simulation model considers a 10 Mbps Ethernet network, where each station has a Poisson traffic source with a fixed packet length of 250 bytes. The total network load ranges from 40% to 110%. For each simulated load,  $75 \times 10^4$  packets are successfully transmitted.

Once more, it is considered a shared Ethernet environment, where multiple stations are interconnected with a *special station* implementing either the h-BEB (enhanced Ethernet mode) or the BEB algorithms (traditional Ethernet mode). Two scenarios are assessed: the small population scenario with 5 Ethernet stations, and the large population scenario with 65 Ethernet stations.

The target of the simulations is to analyze the behavior of the h-BEB algorithm when compared to the traditional BEB collision resolution algorithm. Therefore, the *special station* is used as the test case for both scenarios. The performance measures include both the maximum access delay for 80%, 95%, 98% and 99% of messages and the standard deviation of the average access delay (transfer jitter). The maximum access delay is the maximum time required to successfully transfer a packet, measured from the first transmission attempt to the end of the packet transfer. The maximum access delay for  $x\%$  of the messages is evaluated discarding the  $(100-x)\%$  slowest messages. The standard deviation, which is related to the message transfer jitter, is given by:

$$\sigma = \sqrt{\frac{1}{N} \times \sum_{i=1}^N (x_i - \bar{x})^2} \quad (4)$$

where  $N$  is the total number of simulated packets,  $x_i$  is the delay of each transferred packet and  $\bar{x}$  is the evaluated average packet delay. Discarded packets are not considered for the average packet delay, as this measure deals with just the successfully transferred packets.

### 5.2.1 The small population case

Figures 7 and 8 illustrate the maximum access delay for 80%, 95%, 98% and 99% of the messages and its standard deviation (transfer jitter) in the small population case, for both the enhanced and the traditional Ethernet modes.

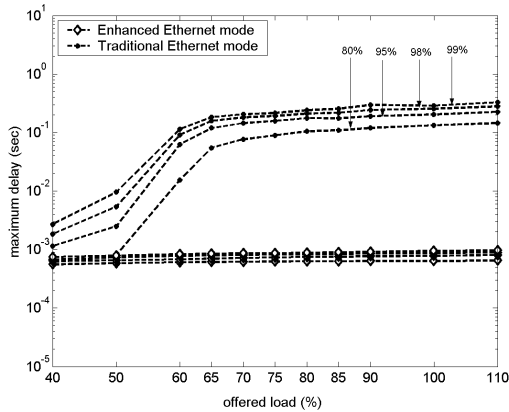


Figure 7: Maximum Access Delay for the small population case.

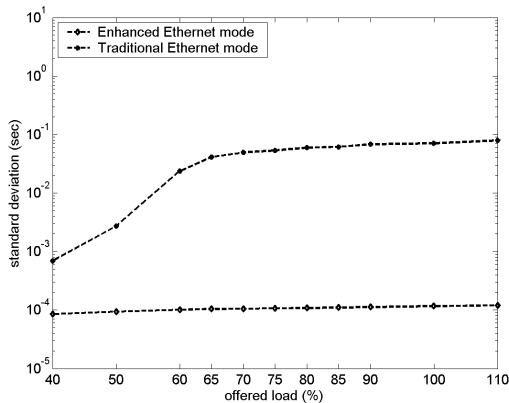


Figure 8: Standard deviation of the average delay for the small population case.

Figure 7 show that the maximum access delay for  $x\%$  of the messages is nearly constant for the enhanced network case scenario, whatever the network load. More importantly, Figure 8 illustrates that the standard deviation of the average delay is one order of magnitude smaller than the maximum access delay for  $x\%$  of the messages, which indicates a small dispersion of the simulated results.

Moreover, as the standard deviation of the average delay is a measure of the message transfer jitter, it becomes clear that, whatever the network load, the enhanced Ethernet mode guarantees a nearly constant message transfer jitter. This is an important result, as it forecasts a predictable communication delay when supporting real-time communications.

Finally, both Figures 7 and 8 clearly illustrate the behavior of the traditional Ethernet mode: high access delays for network loads above 60%, with a standard deviation of the average delay in the same order of magnitude of the maximum access delay; the latter indicates high message transfer jitter.

From Figures 7 and 8 it is not clear that for the traditional Ethernet mode, the packet rejection rate becomes significant for network loads above 60%, while for the enhanced Ethernet mode it was not detected any discarded packet within the  $75 \times 10^4$  simulated transfers.

### 5.2.2 The large population case

Figures 9 and 10 illustrate the maximum access delay for 80%, 95%, 98% and 99% of the messages and its standard deviation (transfer jitter) in the large population case, for both the enhanced and the traditional Ethernet modes.

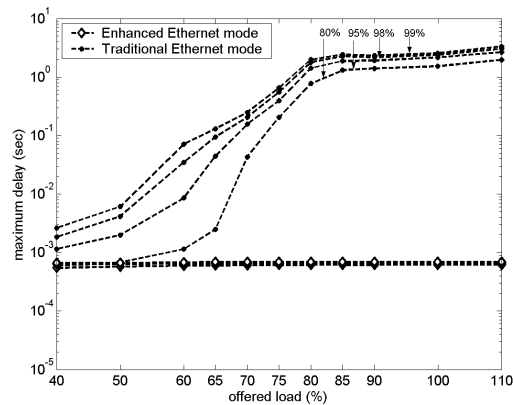


Figure 9: Maximum Access Delay for the large population case.

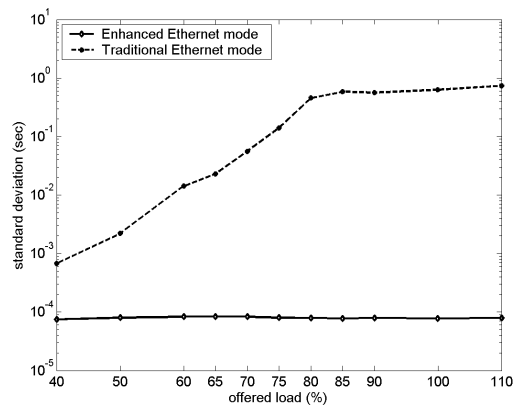


Figure 10: Standard deviation of the average delay for the large population case.

The presented results illustrate that, for the enhanced Ethernet mode, the results are similar for both the large and the small population cases (there is just a slight decrease in the dispersion of the results). Also, it was not detected any discarded packet within the  $75 \times 10^4$  simulated transfers. These results indicate that the h-BEB algorithm behaves well, whatever the number of node stations in the network segment.

On the other hand, the results are clearly worse for the traditional Ethernet mode, when compared to those of the small population case: both the maximum access delay and the standard deviation of the average delay are one order of magnitude higher for network loads above 70%.

## 6. STATE-OF-THE-ART IN INDUSTRIAL ETHERNET

Basically, Ethernet networks went through a significant modification from the shared Ethernet specification [5], when the full-duplex operating mode was introduced in the early 90s (IEEE 802.1D) [10], specifying bridges (also referred as Ethernet Switching Hubs) to interconnect node stations. Such full-duplex operating mode enables the micro-segmentation of the network, by regenerating information only to the receiving port of the bridge, therefore avoiding collisions between messages. Additionally, when using Ethernet Switching Hubs, it is possible to manage network traffic, by means of the adequate setting of data flow permissions and priorities. The transfer of critical information was addressed both by the IEEE 802.1p and the IEEE 802.1q VLAN [11] standards; the latter extends the priority handling aspects of the 802.1p standard, by providing space in the VLAN Tag to indicate traffic priorities to support virtual local area networks (VLANs), while the former gives the ability to prioritize messages.

Nevertheless, the use of switches in an Ethernet network is not a panacea. For instance, if the traffic is sent to an output port at a higher rate than its capacity, messages must be queued. If queuing occurs in an uncontrolled way, the switch can lose messages. Another important problem concerning the use of switched Ethernet is the lack of enough priority levels to support efficient priority-based scheduling [1]. The impact of network topology and message scheduling strategies inside the switch has also been recently addressed [12].

However, numerous Ethernet networks still operate in heterogeneous environments, with Ethernet Switching Hubs interconnecting both independent node stations and Ethernet Repeater Hubs with multiple interconnected node stations (equivalent to shared Ethernet segments). In such heterogeneous environments, the Switching Hubs impose separate collision domains at each port (network segmentation), allowing the implementation of service policies with different priorities. However, within each of the collision domains (*i.e.*, among node stations interconnected by each Repeater Hub), the network still operates in the traditional shared Ethernet mode; that is, collisions are solved by means of a probabilistic contention resolution algorithm, *i.e.*, the medium access is inherently non-deterministic.

Three approaches can be considered to support real-time communications in shared Ethernet

environments [3]: either *avoiding collisions*, by controlling the medium access rights of each station (TDMA scheme, token passing, etc.), or ensuring a *deterministic collision resolution* scheme, by modifying the collision resolution algorithm. A third approach (that is not deterministic) is to reduce the number of occurring collisions, enhancing the network responsiveness to real-time message requests. Whatever the selected approach, it requires the implementation of the protocol modification *in all the interconnected node stations* (at the network adapter level or above), which makes difficult the support of real-time communications within legacy Ethernet communication systems.

The International Electrotechnical Commission (IEC) originally defined three solutions for Industrial Ethernet in the IEC standard 61158. However, there are several systems with potentials to fulfill real-time Ethernet specifications: *Profinet*, *EtherNet/IP*, *EtherCAT*, *Ethernet Powerlink* and *Modbus*, which are briefly summarized in this section.

*Profinet* is the Ethernet-based automation standard maintained by PROFIBUS International and more than 50 companies. In 2003 was ratified as the International Standard IEC 61158 and IEC 61784. According to Feld [13], *Profinet* provides a reaction time in the required range of 5-10 ms for factory automation and, 1 ms and below for motion control applications, which is adequate in terms of real-time responsiveness. In both *Profinet* versions (v2 and v3), a middleware-scheduling layer provides the adequate priority to the real time data. *Profinet v2* can cooperate with IEEE 802.1 compatible network components. The real time channel is based on a cyclic Provider/Consumer architecture, with Ethernet layer 2 frames. *Profinet v2* can support different real-time classes for most application with cycle time in the range of 5 ms and above, using standard switch-based Ethernet technology. However, motion control applications require a cycle time in the range of 1 ms and below, with a jitter in the range of 1  $\mu$ s, impairing the use of switch-based Ethernet technology, especially is standard IP traffic is scheduled in parallel to real-time data [14]. *Profinet v3* is based on TDMA scheduling that supports different real-time classes, and it is also compatible with the IEEE 802.1 standard [14]. The TDMA scheduling is based on a communication ASIC (Application specific Integrated Circuit), where a time slot is exclusively reserved for real-time communication within the communication cycle.

*EtherNet/IP* is an industrial communication standard originally defined by Rockwell, which is supported by ODVA and ControlNet International. It makes use of an open application layer protocol, which is based on Control Information Protocol (CIP) that is used in both *DeviceNet* and *ControlNet*. This topology implements a common set of service at all the network levels, where all the devices organize their data into a common object model. The CIP family of protocols contains a fairly large collection of

commonly defined objects [15]. Ethernet/IP classifies the network nodes by device types and objects are added according to specific functionalities.

*EtherCAT* is an open technology for which IEC standardization is in progress. It sets new standards for real-time performance using twisted pair or fiber optic cable, and it supports line, tree or star topologies. With *EtherCAT*, the data exchange is fully based on a pure hardware machine over a logical ring structure, where a master clock determines the propagation delay. External synchronization is based on the IEEE 1588 standard. *EtherCAT* has different addressing options for different types of communication, optimized for each particular requirements [16].

*Ethernet Powerlink* protocol is based on the standard IEEE 802.3 layers. Deterministic time is achieved by applying a cyclic timing schedule to all the connected nodes. The schedule is divided in isochronous and asynchronous phase. During the isochronous phase, time-critical data is transferred; the asynchronous phase reserves bandwidth for non time-critical data. The node management grants the access to the physical medium via the exchange of an explicit message (token), thereby preventing collisions. The Ethernet Powerlink Standardization Group (EPSG) recommends the use of repeater hubs instead of switching hubs within the real-time domains, to minimize path delay and frame jitter.

*Modbus* protocol, developed by Modicon in 1979, is based on master-slave/client-server communication between devices. It is a protocol that is positioned at level 7 of the OSI model. It defines a simple protocol data unit (PDU) independent of the underlying communication layers. The Modbus messaging communication uses four type of messages: a *Modbus Request* is the message sent on the network by the client to initiate a transaction; a *Modbus Indication* is the request message received on the server side, a *Modbus Response* is the Response message sent by the Server, a *Modbus Confirmation* is the response message received on the client side.

## 7. CONCLUSIONS

This paper presents the timing analysis of an enhanced collision resolution algorithm for shared Ethernet networks: the high priority Binary Exponential Backoff (h-BEB) algorithm. Both the analytical and the simulation timing analysis show that the h-BEB algorithm guarantees a maximum access delay that is significantly smaller than for the standard Ethernet stations.

Two cases were analyzed. Firstly, the analytical study for a heavily loaded network scenario shows that the maximum access delay for 95% of the messages is smaller than 1,86ms. Secondly, for more realistic load scenarios (intermediate load cases), the simulation analysis shows that the maximum access delay for 98% of the messages is always smaller than 1ms. More importantly, it shows a nearly constant

message transfer jitter, which is one order of magnitude smaller than the maximum access delay for 98% of the messages.

Concerning the probability of a message frame being discarded by the h-BEB algorithm, it has also been shown that, for the heavily loaded network scenario, such probability is always smaller than  $2 \times 10^{-3}$ . For more realistic load scenarios, the simulation analysis never detected any discarded frame.

These are important results, as they forecast a predictable communication delay when supporting real-time communications with the h-BEB collision resolution algorithm. These results are also consistent with the claim that the h-BEB algorithm is adequate to support most part of the soft real-time applications, as they confirm a rather small probability of any message being discarded.

## 8. REFERENCES

- [1] J.-D. Decotignie, "A perspective on Ethernet-TCP/IP as a fieldbus," presented at Proceedings of LORIA. 4th International Conference on Fieldbus Systems and their Applications, 15-16 Nov. 2001, Nancy, France, 2002.
- [2] P. Neumann, "Manufacturing Automation over Networks (Keynote Speech)," presented at 11th IFAC Symposium on Information Control Problems in Manufacturing, Salvador - Brazil, 2004.
- [3] R. Moraes and F. Vasques, "A Probabilistic Analysis of Traffic Separation in Shared Ethernet Systems Using the h-BEB Collision Resolution Algorithm," presented at 13th International Conference on Real-Time Systems - RTS'2005, Paris - France, 2005.
- [4] F. Carreiro, R. Moraes, J. A. Fonseca, and F. Vasques, "Real-Time Communication in Unconstrained Shared Ethernet Networks: The Virtual Token-Passing Approach," presented at Emerging Technologies and Factory Automation (submitted to), Catania, 2005.
- [5] "IEEE standards for local area networks: carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications," in ANSI/IEEE Std 802.3-1985, 1985.
- [6] R. M. Metcalfe and D. R. Boggs, "Ethernet: distributed packet switching for local computer networks," *Communications of the ACM*, vol. 19, pp. 395-404, 1976.
- [7] S. S. Lam and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," vol. CM-23, pp. 891-904, 1975.
- [8] G. T. Almes and E. D. Lazowska, "The behavior of Ethernet-like computer communications networks," presented at Proceedings of the Seventh Symposium on Operating Systems Principles, 10-12 Dec. 1979, Pacific Grove, CA, USA, 1979.
- [9] "ns-2 Network Simulator," 2.27 ed, 2004.
- [10] "Information technology - Media access control (MAC) bridges," in ANSI/IEEE Std 802.1D, 1993, pp. 1.
- [11] "IEEE standards for local and metropolitan area networks: virtual bridged local area networks," in IEEE Std 802.1Q-1998, 1999.
- [12] E. Jasperneite and P. Neumann, "Switched Ethernet for factory communication," presented at ETFA 2001. 2001 8th International Conference on Emerging

- Technologies and Factory Automation. Proceedings, 15-18 Oct. 2001, Antibes-Juan les Pins, France, 2001.
- [13] J. Feld, "Realtime Communication in PROFINET V2 and V3 Designed for Industrial Purposes," presented at 5th IFAC International Conference on Fieldbus Systems and their applications, Aveiro, Portugal, 2003.
- [14] J. Feld, "PROFINET - scalable factory communication for all applications," presented at 2004 IEEE International Workshop on Factory Communication Systems. Proceedings, 22-24 Sept. 2004, Vienna, Austria, 2004.
- [15] P. Brooks, "Ethernet/IP - Industrial protocol," presented at 8th International Conference on Emerging Technologies and Factory Automation (ETFA 2001), Oct 15-18 2001, Antibes-Juan les pins, 2001.
- [16] D. Jansen and H. Buttner, "Real-time ethernet the EtherCAT solution," Computing & Control Engineering Journal, vol. 15, pp. 16-21, 2004.
- [17] G. Kaplan, "Ethernet's winning ways," IEEE Spectrum, vol. 38, pp. 113-15, 2001.