# MESSAGE DELAY IN DISTRIBUTED CONTROL SYSTEMS THROUGH ETHERNET

**Juan Gámiz-Caro[1], and Antoni Grau[2]**

[1] *Technical School of Industrial Engineering (EUETIB), Technical University of Catalonia, UPC*
*c/ Urgell 187, 08036-Barcelona, Spain*
juan.gamiz@upc.edu
[2] *Automatic Control Dept (ESAII), Technical University of Catalonia, UPC*
*c/ Pau Gargallo, 5 08028-Barcelona, Spain*
antoni.grau@upc.edu

Abstract: In distributed control system Ethernet is invading communication areas reserved so far to other specific and/or proprietary networks, due to its wide bandwidth, reliability and reduced cost. However this fact obliges to carefully evaluate the delay that messages undergo in order to fulfill with the application temporal requirements. In this paper the delay model is studied, its main components are defined, and, finally, the experimental results and the divergences respect the theoretical values are presented. 

Keywords: Communication network, Ethernet, Communication control applications, Distributed control applications, Model control.

## 1. MESSAGE DELAY IN DISTRIBUTED CONTROL SYSTEMS

The criterion to choose a communications network that fulfills the control requirements in a specific industrial process must take into account the different functional necessities of design, efficiency, cost, maintenance and further upgrade. A common mistake in the choice is to consider the information transfer rate between equipments as a determinant factor. Despite the importance of this factor, other aspects such as the delay in the acquisition of signals and the temporal determinism of the link, must be taken into account.

To forecast the behavior of a communication system time consists, basically, in the analysis what maximum latency times warranty the bus for each message under certain load. The result of this forecast is crucial in critical applications of real time in order to know its feasibility and, so, to allow its scheduling.

Independently of the network, the information transfer through any communication channel requires a time which depends on the bandwidth, the length and overload of the link, the baud rate and efficiency of the network, the electromagnetic disturbances, the number of participating nodes, etc. So, the used time in the communication constitutes a variable and, in many cases, a random delay that adds more complexity to the difficult task of control systems design (Marti *et al*., 2001).

### *1.1. Message delay components.*

Generally, it will be considered that the messages interchanged between nodes are generated with a certain periodicity (*Tm*) and must be delivered in the destination no longer than the deadline, denoted by *Dm*. The message is generated by an appropriate task in the sender node, undergoing any delay (*Jm*) when it is queued to the output. In the output queue, furthermore, it can undergo any temporal blocking (*Tbloq*) due to the time needed to send the previous messages. When the message is transmitted, it will arrive to its destination after a propagation time (*Tm*). In Figure 1 the above mentioned delay components can be seen.

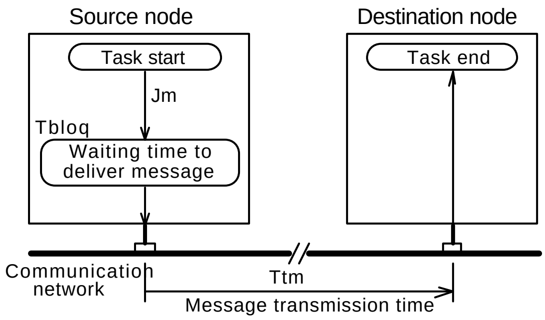


Fig. 1. Message delay components.

The worst response time for a specific message *m*, (Burns *et al*., 1993), (Audsley *et al*., 1993), corresponds to the equation:

$$Tdelay_{(m)} = Jm_{(m)} + Tbloq_{(m)} + Ttm_{(m)} \quad (1)$$

Where *Tdelay* will be considered as the maximum time from that sender node generates the message to it is completely received in the destination node. Thus, the control system will be schedulable if and only if $Tdelay_{(m)} \le Dm_{(m)}$.

## 2. ETHERNET NETWORK

Ethernet (EN 50170, 1996) is a model of local area network, created initially for office automation, with an upward penetration in the area of industrial communications applied to process control (Seok-Kyu *et al*., 1999). The MAC protocol in Ethernet is based on CSMA/CD and probabilistic (Gámiz *et al*., 2003). It is not possible to ensure that the temporal requirements will be fulfilled in all the messages interchanged in this kind of communications network[1]. However, Ethernet is imposing in multiple control systems due, mainly, to its wide bandwidth that warranties a good temporal response when the network is exclusively devoted to a specific control task, with a reduced number of nodes, identical network interfaces and identical length in each node.

[1]Recently, some deterministic variants to the Ethernet network have appeared such as FTT-Ethernet and Switched-Ethernet that can support real-time traffic.

Another aspect that makes Ethernet a valid alternative is its huge diffusion and the large number of low-cost products.

### *2.1. Medium access method in Ethernet.*

CSMA/CD is the protocol of medium access used in Ethernet networks. When a node wants to transmit checks the bus to be idle. If a node does not need to transmit observes the line to know whether the other nodes have begun to transmit any information to it. The node that wants to transmit will wait until the bus is free, setting to zero the collisions counter ($N=0$) for the frame to transmit. The node will wait the *IFG* time (*inter-frame gap*) of 9.6 μs (12 bytes at 10Mbps) to allow the propagation of the signal through the destination receptor electronics, and then, the node will transmit the message. If two or more nodes try to transmit simultaneously a collision is produced corrupting the data.

The procedure used by the transmitting nodes when they try to transmit again their frames due to collisions is dictated by an algorithm named *back-off*. Basically, the message retransmission mechanism is as follows. When a collision is detected, every node will wait a random time before to transmit back its message. The waiting time is calculated multiplying a random number, from a possible set, by the slot time (that is, the minimum frame time: 51.2 μs at 10 Mbps). After the waiting time plus the IFG time, the node tries again the frame transmission. As the retransmission can still collide, the protocol will try it up to 15 times. If the collision persists after the fifteenth retry the transmitter ends the transmission, rejects the frame and an error is reported. For each frame retransmission, the transmitter forms the set: $\{0, 1, 2, 3, 4, 5, ... L\}$, where $L = 2^K - 1$ (being $K$ = collision number, and $K=10$ as a maximum), and it chooses a random value $R$ from this set. The transmitter tries to transmit the frame after a time *R*, the slot time.

## 3. MESSAGE DELAY MODEL IN ETHERNET

In this section the message delay components will be obtained in an Ethernet network as *Ttm* and *Tbloq* times showed (1). In the following analysis the jitter time, *Jm*, will be considered insignificant compared with *Ttm* and *Tbloq* times and, consequently, it will be cancelled.

### *3.1. Message transmission time (Ttm).*

The transmission time of message *m*, $Ttm_{(m)}$, depends on its frame length, $Rlt_{(m)}$, and on the propagation delay due to the line length, *Rpr*, that is:

$$Ttm_{(m)} = Rlt_{(m)} + Rpr \quad (2)$$

If $Ndata_{(m)}$ is the number of useful bytes in the frame of message *m*, *Ncab* the number of bytes in the header, *Nrell* the number of stuff bytes to reach the minimum size and *B* the network baud rate, the delay time $Rlt_{(m)}$ will be:

$$Rlt_{(m)} = \left(Ncab + Nrell + max\left[Ndata_{(m)}, 46\right]\right) \cdot 8 \cdot \frac{1}{B} \quad (3)$$

Where: *Ncab* = 12 bytes (IFG) + 8 bytes (preamble). *Nrell* = 6 bytes (destination) + 6 bytes (source) + 2 bytes (type) + 4 bytes (CRC). *max [*$Ndata_{(m)}$ *, 46]* is the maximum value between the values in the square brackets, with $Ndata_{(m)} \leq 1500$ bytes. *B* is the transmission speed in bit/s.

The propagation delay in the network, *Rpr*, depends on the distance between the source and destination nodes. The propagation delay in a twisted pair line is about ≈ 6 ns/m. Consequently, at a line distance of 2500m and 100m the delay is 15 µs and 0,6 µs, respectively. Thus:

$$Rpr = l_{line} \cdot 6 \cdot 10^{-9} \quad (4)$$

Where: $l_{line}$ is the line length in meters.

Finally, from (3) and (4) results:

$$Ttm_{(m)} = \frac{304 + 8 \cdot max\left[Ndata_{(m)}, 46\right]}{B} + l_{line} \cdot 6 \cdot 10^{-9} \quad (5)$$

*3.2. Message blocking time (Tbloq).*

*Tbloq* time depends on the network protocol and it is the time that have most incidence in Ethernet network performances used for control (Wheelis, 1993). It states an estimation for the required time in the message re-sending after *k* collisions and it can be represented as:

$$Tbloq_{(m)} = \sum_{k=1}^{16} E\{T_k\} \quad (6)$$

It is difficult an exact analysis of expected blocking time (Lian *et al*., 1999). However, the blocking time $Tbloq_{(m)}$, that affects to a message can be evaluated setting three hypothesis validated afterwards through the suitable experiences.

*Hypothesis I*: The collision probability of a message *m* with other messages in the network is highly related with the network occupation at that time, when the "Ethernet capture phenomena is never present (Ramakrishnan and Yang, 1994), and for medium (< 60%) and low network loads (< 10%, common in Ethernet networks used in control). Defining the parameter $Urex_{(m)}$ as the factor of network occupation due to the network use of all the system messages except *m*, the existing correlation between this parameter and the number of collisions that can affect to a message before its transmission can be obtained.

In the case that all the nodes in the net try to transmit in a continuous way ($Tm_{(m)} = Tdelay_{(m)}$) a unique message , the network occupation factor is defined as:

$$Urex_{(m)} = \frac{\sum_{\forall i \in cex(m)} Ttm_i}{\sum_{\forall k \in c(ms)} Ttm_k} \quad (7)$$

Where: $Ttm_i$ is the message *i* transmission time, *cex(m)* is the set of application messages except *m*, and *c(ms)* is the whole set of application messages.

If all the application messages have an identical useful data length, the transmission time, *Ttm*, for all the messages is also identical. In consequence, equation (7) can be now rewritten as:

$$Urex_{(m)} = 1 - \frac{1}{N_M} \quad (8)$$

Where: $N_M$ is the number of nodes in the application.

*Hypothesis II*: The number of consecutive collisions that the message *m* undergoes is related with $Urex_{(m)}$ with a simple probabilistic law. Thus, with $Urex_{(m)}$ values of 0.5, 0.75, 0.80 and 0.90, for example, the probability that the message *m* undergoes a collision before its transmission is 1/2, 3/4, 4/5 and 9/10, respectively. It is important to notice that the above numerators in the fractions (1, 3, 4, and 9) indicate the probable number of consecutive collisions before the message is transmitted. Setting the upper bound of $Urex_{(m)}$ to 0.95, the probable number of consecutive collisions of message *m* can be defined as:

$$Nc_{(m)} = \frac{Urex_{(m)}}{1 - Urex_{(m)}} \quad (9)$$

For systems with an unique message in every node with identical data length and transmitting in a continuous way, equation (9) can be rewritten as:

$$Nc_{(m)} = N_M - 1 \quad (10)$$

Figure 2 shows, according to the above equations, the evolution of the number of collisions for the message *m* respect $Urex_{(m)}$. The graphics is bounded to 16 collisions maximum.

In order to correlate the number of collisions with the blocking time that the message *m* undergoes, it will be fundamental to take into account the following aspects:

- ✓ All the interchanged messages in the nodes must have an identical useful data length.
- ✓ The electronic interfaces of access to media must be identical in every node that means identical speed, features and performance.

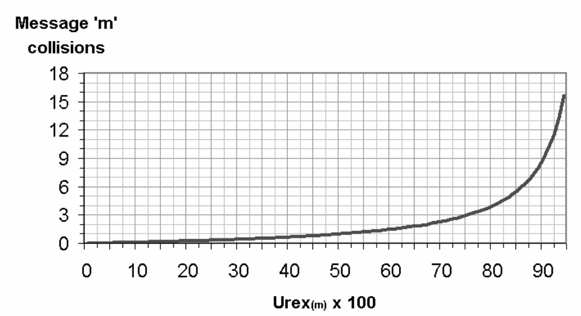


Fig. 2. Number of collisions of message *m* vs $Urex_{(m)}$.

*Hypothesis III.* The blocking time for a message *m* is closely related with the transmission time of the message that it collides with. Consequently, the blocking time of the message *m* can be defined as:

$$Tbloq_{(m)} = Ttm_{(m)} \cdot Nc_{(m)} \qquad (11)$$

The message delay, accordingly to the expression (1), (10) and (11), will be:

$$Tdelay_{(m)} = Ttm_{(m)} \left(Nc_{(m)} + 1\right) = Ttm_{(m)} \cdot N_M \qquad (12)$$

*Relaxation factor 'a'*: This is an important parameter that provides information about the network scheduling where the messages have tight deadline times. It is defined as follows:

$$a = \min_{\forall k \in c(m)} \left( \frac{Dm_k}{Tdelay_k} \right) \qquad (13)$$

Notice that if *a* takes a value equal or less that 1, the network could not fulfill the temporal requirements that messages impose.

### 3.3. 'n' messages in every node and continuous traffic case.

This is the general case for the particular case presented in the last Subsection. In a network with $N_M$ nodes, where all the nodes try to transmit in a continuous way an specific number *n* of messages, the extension of equation (12) is:

$$Tdelay_{(m)} = n \cdot Ttm_{(m)} \cdot N_M \qquad (14)$$

Where: *n* is the number of messages to transmit for every node.

### 3.4. Different number of messages in each node and periodic traffic case.

This case shows a certain periodicity in the message transmission $(Tm_{(m)})$, not necessary the same for all the messages in the system. Furthermore, each message is subjected to a deadline $(Dm_{(m)})$ different. In the same way, every node can hold a different number of messages to transmit. However, all the messages have the same data length.

Such a case is shown in Figure 3. It contemplates a communication system between two nodes (*A* and *B*) with 4 and 2 messages respectively. The messages stack in node *A* is $M_A$, and those in node *B* are $M_B$.

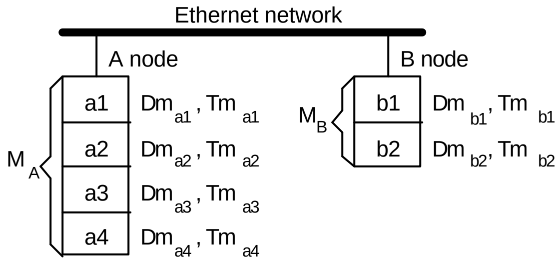


Fig. 3. Nodes with different number of messages subject to different *Dm* and *Tm*.

In such a situation, the reduction of equation (10) can not be applied, using instead equation (7). Now, the network occupation factor is defined as:

$$Urex_{(M)} = \frac{\sum_{\forall M \in cex(M)} \sum_{i=1}^{n} \frac{Ttm_i}{Tm_i}}{\sum_{\forall M \in c(Ms)} \sum_{i=1}^{n} \frac{Ttm_i}{Tm_i}} \qquad (15)$$

Where: *cex(M)* is the set of messages stacks in the application except *M*, *c(Ms)* is the total set of messages stacks in the application, *n* is the number of messages in each node and *Tm(i)* is the transmission period for message *i*.

In consequence, a message *m* belonging to a stack *M* will undergo a delay of

$$Tdelay_{(m)} = n_{(M)} \cdot Ttm_{(m)} \left( \frac{Urex_{(M)}}{1 - Urex_{(M)}} + 1 \right) \qquad (16)$$

Where: $n_{(M)}$ is the number of periodical messages that form the stack *M*.

## 4. EXPERIMENTAL RESULTS

In order to carry out the experiments of communication through an Ethernet network an assay platform has been set up with a star-structure set of nodes (compatible Personal Computers), by means of a hub, with 5-meter segments plugged with RJ-45. The different nodes have been considered as elements of an hypothetical control system (sensors, actuators, controllers, etc.) interchanging information through the Ethernet communication network.

The work procedure has been based in sending frames of identical length from one node to another with several network loads. In the experiments, 1500- and 100-byte frames plus 20 additional bytes have been sent (IFG+preamble).

In this way, whereas a transmitter node sends frames to the receiver, the data flux will be interfered by the information that other nodes interchange in the net. Acting in this manner, when the number of transmitter nodes increases the delay that the frames undergo in the receiver node could be determined.

The network environment has been configured implementing the NetBEUI *(NetBIOS Extended User Interface)* network protocol in each node with the objective of concentrating the traffic in the net inside the Link Layer and Transport Layer services.

Apart of Ethernet, the use of any protocol, i.e. NetBEUI, provokes a slightly different frame traffic with control information. To obtain experimental results derived exclusively from Ethernet frames traffic, the *CommView* software, TamoSoft Inc. (Web 1), has been installed in the destination nodes in order to filter and analyze the interesting frames. Another application installed in these nodes is *DU Meter*, Hagel Technologies (Web 2). This software allows obtaining and plotting the speed in a network in a specific communication session, showing several temporal parameters.

In order to perform the different experiments the authors have developed the *Tranethe* software (Web 3). This application is installed in the transmitter nodes and provides a function that allows sending frames (continuous or periodical) at any moment by a command sent through the network to all the transmitter nodes. The test ends when the receiver nodes have enough information to evaluate the delay.

*4.1. Experiments summary.*

Table 1 and Table 2 summarize the results of different tests with 1500- and 100-byte frames plus a 20-byte header (IFG + preamble). In the far-right column the message delay time (*Tdelay*) obtained from equation (16) with $n_{(M)}=1$ is presented.

After the analysis of the values shown in the tables 1 and 2 it is remarkable the concordance between the experimental values *Tdelay* and theoretical values for a given baud rate, and under different network loads.

Figure 4 shows the evolution of *Tdelay* time for a 1500-byte frame versus the network occupation factor and, in Figure 5, the error respect its theoretical value evaluated with equation (16).

Table 1. Summary of experimental results for frames of 1500 bytes + 20 bytes (IFG + preamble).

| Baud rate (Mbit/s) | Urex (x 100) | Experimental Tdelay (ms) | Theoretic Tdelay (ms) |
|---|---|---|---|
| 9.05 | 7.98 | 1.461 | 1.459 |
| 9.19 | 14.68 | 1.552 | 1.551 |
| 9.08 | 26.64 | 1.826 | 1.825 |
| 9.16 | 31.83 | 1.947 | 1.946 |
| 9.22 | 42.98 | 2.313 | 2.312 |
| 8.93 | 53.78 | 2.944 | 2.945 |
| 9.29 | 63.85 | 3.621 | 3.621 |
| 9.19 | 68.01 | 4.129 | 4.135 |
| 9.29 | 73.29 | 4.900 | 4.899 |
| 8.83 | 80.15 | 6.907 | 6.937 |
| 8.94 | 90.78 | 14.758 | 14.751 |
| 8.86 | 94.51 | 24.981 | 24.998 |

Table 2. Summary of experimental results for frames of 100 bytes + 20 bytes (IFG + preamble).

| Baud rate (Kbit/s) | Urex (x100) | Experimental Tdelay (ms) | Theoretic Tdelay (ms) |
|---|---|---|---|
| 633.30 | 1.46 | 1.538 | 1.537 |
| 652.00 | 3.34 | 1.524 | 1.522 |
| 735.70 | 15.59 | 1.524 | 1.545 |
| 755.60 | 16.65 | 1.524 | 1.523 |
| 1070.00 | 42.24 | 1.552 | 1.553 |
| 1080.00 | 54.87 | 1.949 | 1.969 |
| 1760.00 | 67.56 | 1.679 | 1.681 |
| 2290.00 | 77.15 | 1.820 | 1.834 |
| 2450.00 | 88.51 | 3.408 | 3.410 |

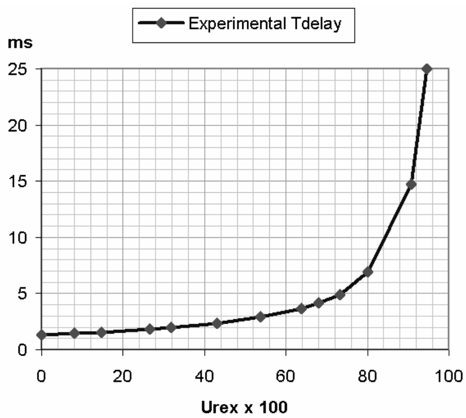


Fig. 4. Tdelay vs Urex in frames transmission of 1500 bytes + 20 bytes (IFG + preamble).

The curve in Figure 4 looks like the curve shown in Figure 2 comparing, first, the close relationship between the collisions number in Figure 2 and the blocking time that the messages undergo in Figure 4 (Y axis) and, second, between the blocking time that the messages undergo and the network occupation factor in Figure 4 (X axis).

Figure 5 confirms the hypothesis set up in Section 3 and also the validity of those equations. The worst case shows a blocking time of *Tbloq* = 7 ms with a network occupation factor of 0.8, and the difference respect the theoretic value is merely 30 μs, that is, a relative error of 0.043%.

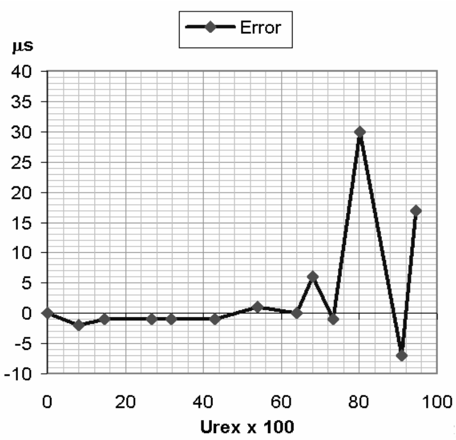


Fig. 5. Difference between theoretic and experimental Tdelay vs Urex in frames transmission of 1500 bytes + 20 bytes (IFG + preamble).

## 5. CONCLUSIONS

In this paper the delay that a message undergoes in an Ethernet network for distributed control systems has been developed. Even though Ethernet is not considered a deterministic network and, consequently, its use in real-time control systems can be conditioned by temporal requirements imposed by the messages, in many control applications is implanted as a solid alternative. Due to its wide bandwidth providing a good temporal response when the network is devoted to an exclusive specific control task, Ethernet has invaded areas of process control reserved so far to other networks and field bus with expensive elements and equipment.

From the general model of message delay in a communications network, and justifying the different hypothesis, the components of Ethernet message delay have been defined in the cases of: one message in each node, *n* messages in each node and different messages number in each node, in applications with continuous as well as periodical traffic. In the same way, the network relaxation factor ($\alpha$) has been defined, providing a useful indicator to know whether the communications system can be scheduled or not.

The different tests carried out revealed the feasibility of the presented model, because the theoretical calculated values support significantly the values obtained in the tests. These results are, also, fruit of many sessions of acquisition and data checks where, besides, the effects of collateral delays (due to the used hardware and software tools) have been minimized.

## REFERENCES

Audsley, N., A. Burns, M. Richardson, K. Tindell, A. Wellings (1993). Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling. *Software Engineering Journal* **8**(4.15) pp. 284-292.

Burns, A., M. Nicholson, K. Tindell, *et al*. (1993). Allocating and Scheduling Hard Real-Time Tasks on a Point-to-Point Distributed System. *Proc. of the The Workshop on Parallel and Distributed Real-Time Systems*, pp. 11-20, Newport Beach, California.

EN 50170 (1996). General Purpose Field Communication System. *Vol. 1 - P-NET, Vol. 2 - ETHERNET, Vol. 3 - WorldFIP*.

Gamiz, J., H. Martinez, J. Domingo, A. Grau (2003). *Communications in the industrial environment*, UOC Editions, 1st ed., Barcelona, 2003, (*spanish version)*.

Lian, F.L., J. Moyne, D.M. Tilbury (1999) *Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet*. Tech. Rep. UM-MEAM-99-02.

Marti, P., J.M. Fuertes, G. Fohler (2001). An Integrated Approach to Real-Time Distributed Control Systems Over Fieldbuses. *8th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, France, pp.72-78.

Ramakrishnan, K.K., H. Yang (1994). The Ethernet capture effect: Analysis and solution. *Proc. 19th Conf. Local Comp. Networks*, Minneapolis, MN, pp. 228-240.

Seok-Kyu Kweon, Kang G. Shin, Qin Zheng (1999). Statistical Real-Time Communication over Ethernet for Manufacturing Automation Systems. *Proc. IEEE Real-Time Technology and Applications Symposium*, pp. 192-202.

Wheelis, J.D. (1993). Process control communications: Token Bus, CSMA/CD, or Token Ring?. *ISA Trans*. **32** (2) pp. 193-198.

Web1. http://www.tamos.com/products/commview/

Web2. http://www.dumeter.com/

Web3. http://webesaii.upc.es/doct/ceib/tranethe.exe