Scheduling Issues in Real Time Systems for Industrial Network

Mario Collotta [1-2], Roberto Di Natale [1-3], Antonio Intagliata [1-3], Angelo Sciria [1-3]

Real-Time Systems and Networks Laboratory (1)

Department of Computer Engineering and Telecommunications - University of Catania (2)

Kore University – Enna (3)

ITALY

mcollotta@diit.unict.it - robertodinatale@interfree.it - antonio.intagliata@alice.it - angelosciria@gmail.com

Abstract — One aspect to investigate in the attempt to enhance the support of process control traffic is the local transmission scheduling algorithm used to queue the requests. In the factory automation the correctness of many systems and devices depends not only on the effects of results they produce, but also on the time at which these results are produced. This paper presents one deadline aware scheduling approach to transmit real-time periodic and aperiodic requests and non-real-time requests.

Keywords: Industrial Automation, Process Control, Real-Time Scheduling, Earliest Deadline First - EDF, Total Bandwith Server – TBS, IEEE 802.11.

I. Introduction

In industrial automation scenarios and process control applications increasing use has been made of wireless communication devices instead of, or combined with, traditional wired connections [1]. The issues of the process control network are the correct management of real-time constraints. In this sense one of the aspects that should be analyzed is the delay end-to-end. In our work we show the real-time performance in an industrial scenario using the standard protocols Ethernet [2] and IEEE 802.11[3].

The measure of response time end-to-end is affected by:

- local queue latency time (LQLT)
- transmission time delay (TTD)

The LQLT is the difference between ready to transmission time and send time. Real-time systems should manage the node local scheduling because if it is not a deadline aware scheduling, performance may be compromised. In this sense local scheduling needs to be improved.

The TTD is the difference between arrival time and send time.

It is important to analyze the real-time aspects offered by the transmission protocol.

This paper mainly focuses on the first aspect, and shows improvements to manage real-time requests, in our work we measure the real-time performances (i.e. the response time end-to-end and the LQLT) .

II. Real-Time Overview

The most important feature of a real-time system is to respond immediately to a real-time request in according to the deadline values. As a result, the scheduler of a real-time system must support a priority based algorithm. Moreover the priority-based scheduling algorithm assigns each request a priority depending on its importance [4].

After the "deadline" has passed, results may be useless. In Hard Real Time Systems, missing the deadline may be catastrofic; they must guarantee that real-time tasks are serviced within their deadline; in Soft-Real-Time Systems performance degrades in case of deadline misses, but they may be tolerated. Soft-RT systems are the least restrictive, assigning real-time requests higher scheduling priority than other requests. They usually are general purpose. Generally, a RT request is characterized by some important parameters:

- arrival time (a): instant of time when a request is ready to be transmit ;
- duration time (C): it is message duration time of the request;
- deadline (d): it is instant of time in which the request must be received;
- start time (s): it is instant of time when the request is sent;
- finishing time (f): it is instant of time when the request is received;

In industrial scenarios there are three types of real-time requests. The aperiodic request is one shot type. The periodic request has many instances or iterations, and a fixed period exists between two consecutive transmission of the same request type. A sporadic request has zero or more instances, and a minimum interval occurs between two consecutive transmission of the same request type. Therefore the periodic requests and sporadic requests are defined respectively also

by the period and the minimum interarrival time. All the parameter types shown in this paper are considered in the worst case. For example we use the worst-case transmission (or message duration) time (WCTT) as C. This time (C) is not simply an upper bound on the transmission of the request. In [5] it shows the importance to determine the computation time of a process (in a CPU-scheduling) or the message duration (in a message scheduling), is crucial to successfully schedule requests or processes in a real-time system.
Two important scheduling algorithms in a real-time system are Earliest Deadline First (EDF) [6] to manage periodic requests and Total Bandwidth Server (TBS) [7] [8] to manage aperiodic and sporadic request.

A . EARLIEST DEADLINE FIRST

EDF [6] is an algorithm that assigns every request a dynamic-priority, inversely proportional to its relative deadline (D): it means that at the first time (when r_i is equal to zero) request is ready to run, it is assigned a D_i value equals to its d_i; the relative deadline starts to decrease by the following equation:

$$D_i = d_i - r_i$$

where r_i is the time elapsed from the instant in which the request enters in the queue of ready-to-run requests.
Usually in a real real-time scenario [9] the d_i value is equal to period value (T_i). EDF is able to schedule also requests that are not necessary periodic, this is the reason why this algorithm is largely used in general purpose systems.

B. TOTAL BANDWIDTH SERVER

TBS, proposed in [7][8], is an algorithm serving aperiodic and sporadic requests in a dynamic context. In the paper, to simplify the name utilized, we call the aperiodic and sporadic request with the same name: aperiodic request. Nevertheless we leave their different meanings.
The main advantages of the TBS algorithm are simplicity of implementation and low computing costs. According to the TBS algorithm, when an aperiodic request arrives at time $t=r_k$, it receives a deadline d_k which depends on the bandwidth allocated to previous aperiodic variable transmission requests, the transmission time of the request itself and the server bandwidth, according to the following rule:

$$d_k = max(r_k, d_{k-1}) + \frac{C_k}{U_s} \qquad (1)$$

where $\max(r_k,d_{k-1})$ is the maximum value between the arrival time of the request and the deadline (d_{k-1}) of the previous one in the queue, C_i the message duration time of the request and U_s value is the amount of the bandwidth utilization of the Server. It is the percentage of bandwidth that needs to send the aperiodic messages. After the deadline is assigned, the request is ready to enter in the scheduling queue and to be scheduled with EDF algorithm.

III. THE NOVEL SCHEDULING APPROACH

As in [10][11], we assume that traffic exchanges are known a priori, at least for periodic variables, this is typical of an industrial communication scenario.
The behavior of a variable depends on whether it is periodic or aperiodic/sporadic. If it is periodic, the period T specifies a constant interval between invocations. If it is sporadic, we define the minimum interarrival time between invocations, which we will indicate as T_m. If it is aperiodic, we should have a one shot request.
Periodic requests are served according to EDF. Based on the knowledge of the dynamics of the periodic messages produced by the various Clients. Aperiodic traffic is handled in the same way as sporadic traffic. The TBS algorithm is handled by the Server to schedule the aperiodic and sporadic requests. Whereas the non real-time request is handled in background mode.
To properly management of the Server scheduling, we will use a schedulability theorem that derives from the one described in [12], but it is adapted to our case.
A set of periodic requests can be scheduled using the non-preemptive EDF algorithm if two conditions are met. The first (2.1) relates to system utilization (in terms of bandwidth, as we are dealing with the transmission of packets), whereas the second (2. 2) is a least upper bound on the demand for bandwidth that can be achieved in an interval of length L, starting from the instant at which the periodic variable is invoked and ending before the relative deadline.

Theorem : *Let us consider a set of periodic variables τ_p ={v1, v2, ..,vn}, where vi = (ci, Ti), sorted in non-decreasing order by period (i.e., for any pair of variables v_i and v_j , if $i > j$ then $T_i \geq T_j$). If τ_p is schedulable then:*

$$U_{tot} = U_p + U_s = \left(\sum_{i=1}^{n} \frac{C_i}{T_i} + U_s \right) \leq 1 \qquad (2.1)$$

$$1 < i \leq n; \forall L, T_1 < L < T_i : L \geq C_i + \sum_{j=1}^{i-1} \left\lfloor \frac{L-1}{T_j} \right\rfloor c_{Cj} \qquad (2.2)$$

where C_i is the transmission time for a periodic variable.

The first inequality, shows that the total bandwidth utilization must not exceed 1, U_p is the periodic utilization bandwith it refers to periodic traffic, U_s value is the utilization bandwith, it refers to sporadic/aperiodic, it is the utilization of the TBS server.

IV. TESTBED

To test the efficiency of the proposed approach, a client-server system was created, in which the server handles the client's requests using EDF and TBS Scheduling, developed in Linux (kernel 2.6.27-11) environment using the 'C' language. Server Node (SN) waits for all the Client Nodes (CN) to connect and it is ready to accept every kind of CNs' request.

The following figure shows the request structure.

```
struct request
{
 char name [1024];
 int type;
   //if type=0 the request is periodic
   //if type=1 the request is aperiodic
   //if type=2 the request is sporadic
   //if type=3 the request is non-real-time

 float T;
   // If the request is periodic it is the period
   // If the request is sporadic, it is the minimum interarrival time

 float C;
   // duration Time  of real-time message

 float deadline;
   // if the request is periodic it is Absolute Deadline
   // if the request is sporadic or aperiodic it is  deadline value assigned by TBS algorithm
};
```

Figure 1 : request format

All requests are associated also some information about:

- Queue-time: this is the time that the request is in queue of ready-to run requests;
- The value of relative deadline when it has been executed;
- The flag that indicates the "deadline miss" or not;

We examined two different scenarios, with standard scheduling and with the deadline-aware scheduling here-in described. In both cases using respectively Ethernet and 802.11.
The following figure shows the first scenario, where there are 5 CNs, they are linked at the SN by wired Ethernet protocol

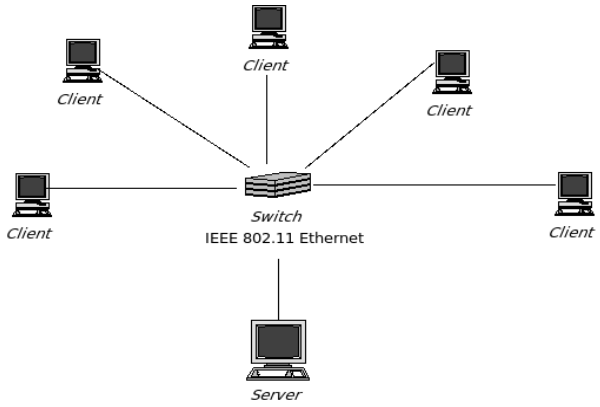

Figure 2: Clients/Server Scenario through a Ethernet Switch

The figure 3 shows second scenario where the CNs of the network communicate using a common Wireless Access Point in a IEEE 802.11 network. It links five CNs with a Server Node.

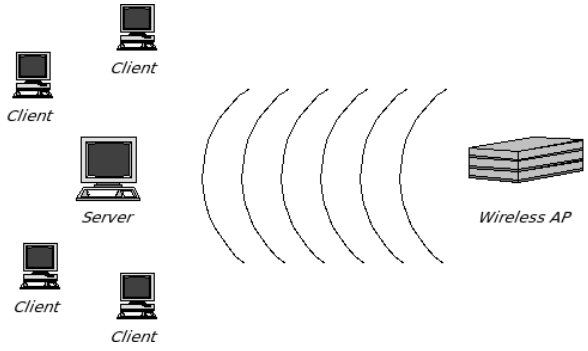

Figure 3: Clients – Server through a Wireless AP

V. Performance Evaluation

This paragraph shows performance evaluation of the simulation. The simulation consists in bursts of periodic, sporadic and non-real-time requests.
In our testing periodic requests are three different burst types and they flood the network. They have the values shown in table I.

Periodic Requests	Deadline/Period	C_i
R0	8	1
R1	10	2
R2	25	4

Table I : Periodic burst specification

The workload of the network, simulated in this paper, is shown in table II.

	Workload
RT-Periodic Request	70 %
RT-Sporadic Request	15 %
RT-Aperiodic Request	5%
NRT Request	10 %

Table II : Workload

In both scenarios we sent the periodic messages only if their deadline is not miss. In this sense we measured the benefits in terms of deadline constraints and we obtained a deadline miss value, using our deadline-aware approach, equal to 0%, while using standard scheduling the deadline miss value is equal to 65% in the first scenario (Fig. 2) and 75% in the second scenario (Fig. 3)
In the following paper section we show the evaluation on end-to-end delay (L) for aperiodic and sporadic real-time requests and non real-time requests. It was calculated using the following equation:

$$L = l\_network + l\_queue + C_i \qquad (3)$$

where:

- l_network, called in the first paragraph TTD (transmission time delay), it is the difference between

arrival time and send time. It depends by network implementation.

- The l_queue value is local queue latency time, called LQLT, it is the difference between ready to transmission time and send time. It depends by the local queue scheduling.
- C_i is the duration time of the i-esimo real-time message.

In the following figures we show the latency values (l_network), they depend by network implementation (i.e. MAC protocol used) in which the system was tested. The average latency of Ethernet network (fig.2), using a switch, through network cables is 0.0005 seconds.

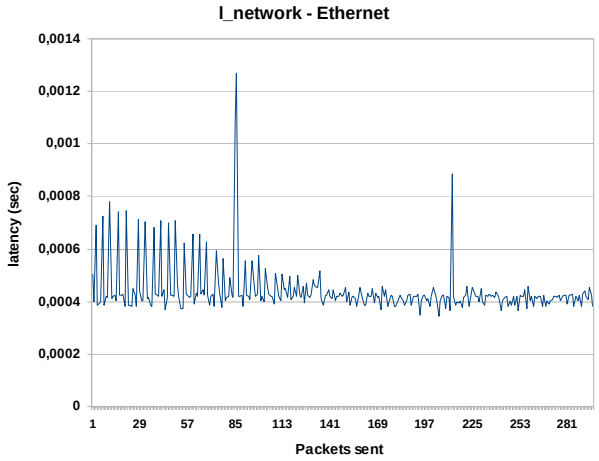

Figure 4: Latency using a Switch

The average latency of the network, fig. 3, using a wireless access point is 0.15 seconds. In the wireless cases the latency is more high than previous case because in the wirelss network it is necessary to improve the physical access mechanism, to avoid that real-time performance may be compromise. A deadline aware scheduling is non sufficient for the correct working of the system, it is necessary but not sufficient.

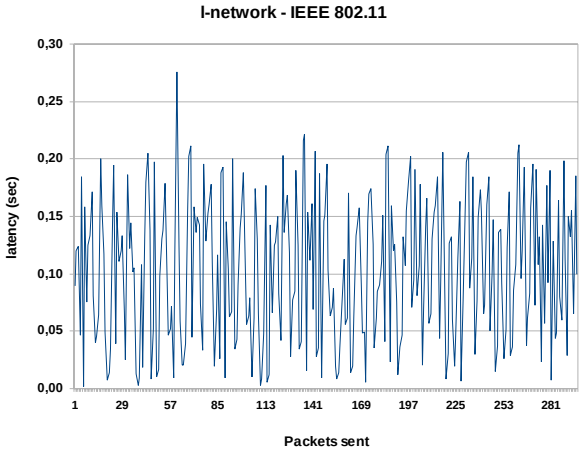

Figure 5: Latency using a Wireless Access Point

The following figures show the l_queue values; it means the time elapsed in ready-to-run queue of sporadic and non-real-time requests.

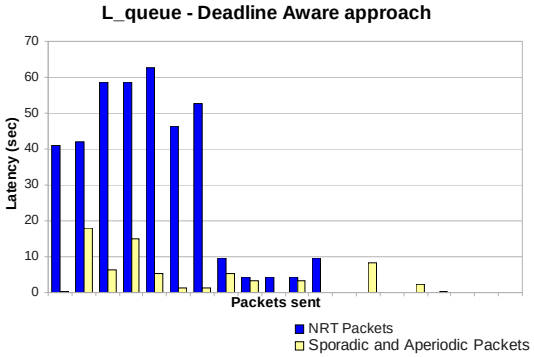

Figure 6: L_queue – Deadline Aware approach

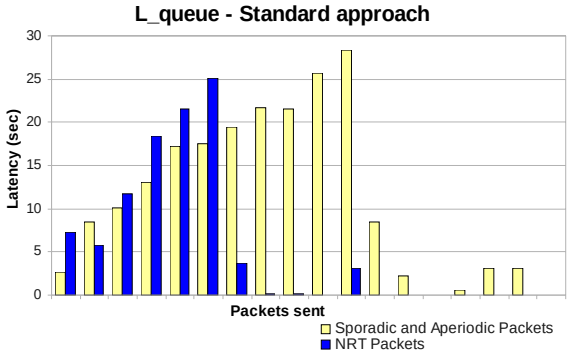

Figure 7: L_queue – Standard approach

It can observe the improving of performance in case of Deadline aware scheduling, in-fact a sporadic and aperiodic requests remain in queue less time than standard case.

The figures 8 and 9 show the end-to-end delay respectively in deadline aware approach and in standard approach, using Ethernet protocol.

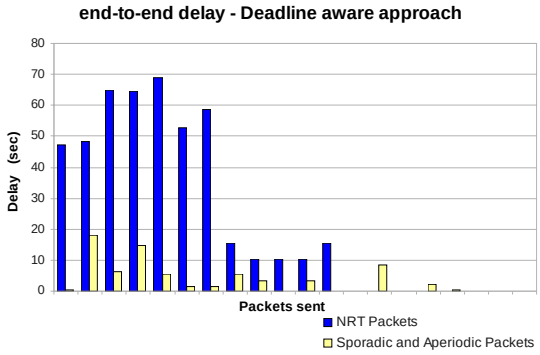

Figure 8: end-to end delay – Deadline aware approach

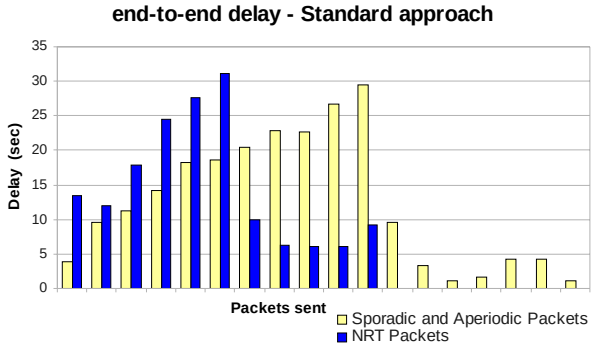

Figure 9: end-to end delay– standard approach

In the following it's shown the performance in terms of end-to-end delay of sporadic and aperiodic packets, measured in wireless scenario (second scenario here described) using 802.11 network.
So in this case the latency is lower using the novel approach shown in this paper, than in standard 802.11 approach.

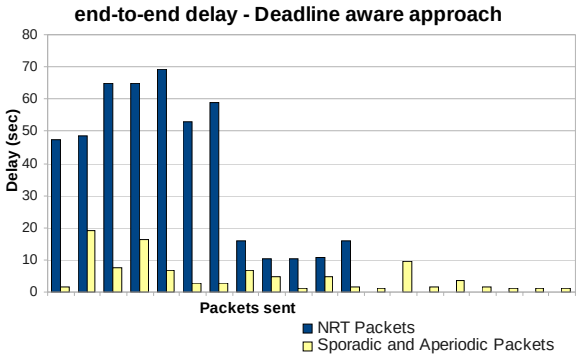

Figure 10: End-to-end delay - deadline aware approach.

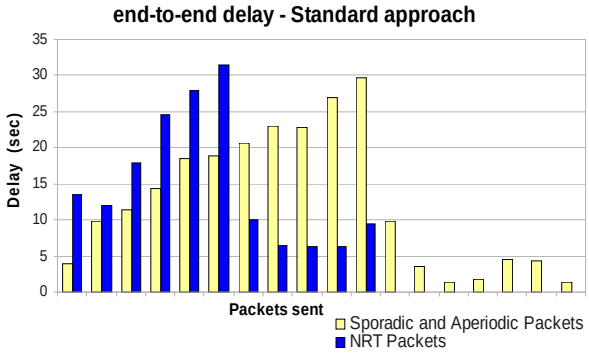

Figure 11: End-to-end delay- Standard approach.

VI. Conclusion

This paper shows that the efficiency of a real-time system for industrial networks depends on the local scheduling algorithms. The described approach, to real-time scheduling in process control networks proposed in this paper, showed the management of bandwidth available for periodic, sporadic and aperiodic transmissions. The approach also has the advantage of handling periodic, sporadic and aperiodic traffic in an integrated fashion. The use of deadline-aware scheduling techniques, i.e. EDF and TBS, allows periodic traffic deadlines to be met in the various scenarios examined, while in the same conditions the standard Ethernet and 802.11 experiences significant periodic packet loss, deadline miss values and very high delay end-to-end values.

References

[1] Gianluca Cena, Ivan Cibrario Bertolotti, Member, IEEE, Adriano Valenzano, and Claudio Zunino, Evaluation of Response Times in Industrial WLANs , IEEE transactions on industrial Informatics, Vol. 3, NO. 3, August 2007.

[2] C.Venkatramani and T. Chiueh "Supporting Real-Time traffic on Ethernet", Proceeding of 15 th of Real-Time Systems Symposium, pp 282,286 – 1994.

[3] IEEE 802.11, Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Standard, IEEE, Aug. 1999.

[4] Albert M. K. Cheng, "Real-Time Systems – Scheduling, Analysis and Verification", Wiley, *2002.*

[5] Silberschatz, Galvin, Gagne, "Operating System Concepts, 7th Edition", Wiley, 2005.

[6] C. L. Liu and J. W. Layland. "Scheduling Algoritms for Multiprogramming in a Hard-Real-Time Environment", Journal of ACM, Vol. 20, No. 1, January 1973.

[7] M. Spuri, G. C. Buttazzo. Efficient Aperiodic Service under Earliest Deadline Scheduling. In Proceedings of the 15th IEEE Real-Time Systems Symposium, San Juan, Puerto Rico, 1994.

[8] M. Spuri, G. C. Buttazzo. Scheduling Aperiodic Tasks in Dynamic Priority Systems. Journal of Real-Time Systems, Vol. 10, No. 2, pp. 1-32, 1996.

[9] G.,Buttazzo, "Hard Real-Time computing systems: Predictable Scheduling Algorithms and Applications", Second Edition Springer (2005)

[10] U Bilstrup, P.A. Wiberg, Wireless Technology in Industry: Applications and User Scenarios, Proc. IEEE ETFA 2001, IEEE Conference on Emerging Technologies in Factory Automation, October 2001, Antibes, France.

[11] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," JSAC, 18, 2000, pp.535–547.

[12] K. Jeffay, D.F. Stanat, C.U. Martel. On Non-Preemptive Scheduling of periodic and Sporadic Task. In Proceedings of the 12th IEEE Real Time System Symposium, pp.129-130, San Antonio, TX Dec. 1991.