

PRIORITY INVERSIONS IN RATE MONOTONIC 8802/4: ANALYSIS OF THE WORST CASE

R. Cayssials, J. Orozco, J. Santos and E. Ferro

Dep. Ingeniería Eléctrica, Instituto de Ciencias e Ingeniería de Computación
Universidad Nacional del Sur, 8000 Bahía Blanca, Bs. As., Argentina

E-mail: {iecayss, ieorozco, iesantos, ieferro}@criba.edu.ar

FAX: 00 54 91 553122

I. Introduction

The ISO 8802/4 (Token-Bus) Standard offers an optional priority mechanism intended to operate in a real-time environment, in which messages generated at one node must be transmitted before a certain deadline. Although 8802/4 is able to implement a Fair Round-Robin priority discipline [2, 7], it can not implement a strict Rate Monotonic (RM) discipline because, in certain circumstances, *priority inversions* take place. In those cases, the use of the transmission medium is granted not to the node with the highest priority but to some lower priority node. In order to determine the LAN's schedulability it is necessary to know the worst case of priority inversions that each node may suffer. The purpose of this paper is to present a formally proved set of lemmas and theorems that, applied to a Real-Time Token-Bus LAN operating under the *Rate Monotonic* priority discipline, allows the determination of the number of inversions in the worst case and, therefore, its schedulability.

II. The Empty-Slots Method

Several methods to determine the schedulability of a real-time system have been proposed [3, 4, 5, 8]. In this paper the *empty-slots method* [8] will be used.

$S(m)$ denotes a network of m nodes, named 1, 2, ..., m . In the empty-slots method, time is assumed to be slotted. During the duration of one slot, called *slot-time* and symbolized T_s , the transmission medium is allocated to one and only one node with no preemptions allowed during the slot-time.

Slots are notated t and numbered 1, 2, For the schedulability calculations, the slot-time is taken as unit of time. When all pending messages have been transmitted and no new messages are generated, empty-slots appear. $e_{j(m)}$ denotes de j^{th} empty-slot in the execution of $S(m)$.

In [6] it has been proved that a system $S(m)$ in which nodes may suffer up to I inversions is RM schedulable iff

$$\forall i \in \{2, \dots, m-I\} \quad T_i \geq e_{(I+1)(i-1)}$$

where $e_{(I+1)(i-1)}$ denotes the $(I+1)^{\text{th}}$ empty slot in $S(i-1)$. It is proved to be

$$e_{(I+1)(i-1)} = \text{least } t | t = 1 + I + \sum_{h=1}^{i-1} \left\lceil \frac{t}{T_h} \right\rceil$$

where $\lceil \cdot \rceil$ symbolizes the monadic ceiling operator. In the scheduling analysis that follows, $t_0(i)$ denotes the instant at which the token abandons node i and a simultaneous generation of messages takes place at node i and all downstream nodes following a round without transmissions.

III. The Token-Bus Priority Mechanism

The right of access to the physical medium is given by the possession of a token which is passed by stations residing in the medium. As the token is passed from a predecessor station to its successor, a virtual ring is formed on the real bus.

The 8802/4 priority mechanism belongs to the type that can not implement a strict RMS [9]: It is obvious, for instance, that if a simultaneous generation of messages of different priority takes place after a round with no transmissions, the use of the medium will be randomly granted to the first node receiving the token irrespective of its priority.

The Standard [1] provides an optional mechanism with four levels of priority, called *access classes*. In each station there are four request queues corresponding to the four priority levels; the bandwidth is allocated by timing the rotation of the token around the ring. Messages of the higher priority (class_6) at any station are transmitted always with the only proviso that a certain time, called *High Priority Token Holding Time, HPTHT*, is not exceeded.

Each of the three lower classes is assigned a "*Target*" *Token Rotation Time*, symbolized *TTRT*. For each class the station measures the time it takes the token to circulate around the logical ring. If it returns in less than the target time, the station can transmit messages of that class until the time has expired. Everything happens as if high priority stations had an infinite *TTRT*.

When the token arrives in the node, its *TTRT* is loaded in a counter and a regressive count is started. The node will be able to transmit if at the next token arrival the counter content is not zero. Since the unit of time is the slot-time and one message is transmitted in each slot, to talk about units of time or about messages heard by the node since the last token reception amounts to the same thing. $R(i)$ and $\tilde{R}(i)$ shall denote the *TTRT* of node i and the $\max_{i < k \leq m} \{R(k) | R(k) < R(i)\}$, respectively.

Messages of the same priority within one real (physical) station may be considered to be generated at one *virtual node*. Therefore, a real station can have up to four virtual nodes. In what follows, the number of virtual nodes in the system will be taken as its number of nodes.

IV. Rate Monotonic Scheduling

RM scheduling is a *de facto* standard, supported, among others, by the USA Department of Defense. In it, conflicts of access to the medium are solved by prioritizing the nodes according to monotonically increasing periods. Some additional rule must be provided in order to break ties when nodes have the same period.

The proof of necessity of the ordering rules as well as the formal proof of all the lemmas and theorems that follow are omitted in this summary but will be included in the full paper.

In what follows let us consider a Token-Bus LAN in steady state assembled under the general assumptions described at the Introduction for the empty-slots method, plus the following particular ones:

- a) If a station contains a virtual node belonging to class_6, no other virtual nodes are allowed in it.
- b) Stations containing nodes belonging to lower classes may have up to three virtual nodes, provided the following ordering rules are fulfilled.

A. Ordering Rules

Rule 1: Nodes are placed in the logical ring ordered, in the token sense of rotation (e.g. clockwise), by monotonically decreasing *TTRTs*.

Rule 2: If $T_i < T_j$, $R(i) \geq R(j)$.

Rule 3: From Rules 1 and 2, it follows that if node $_i$ precedes node $_j$ and $T_i = T_j$, then $R(i) \geq R(j)$.

Monotonically increasing periods are therefore translated into monotonically decreasing *TTRTs*. Since the 802.4 priority mechanism is supposed to associate higher priorities to larger *TTRTs*, the mechanism would implement RMS. The tie between nodes with the same *TTRT* is

broken by their placement in the ring: a node preceding another one in the ring is supposed also to have precedence in the right to transmit.

For a given node $_i$, downstream nodes are $i+1$ to m , and upstream nodes, 1 to $i-1$. When in spite of node $_i$ having a queued message, downstream nodes transmit before it, inversions take place. For nodes belonging to the lower classes, inversions may come from nodes having the same TTRT or from nodes having smaller TTRTs. They will be called placement and token inversions, respectively. Nodes belonging to the upper class do not have associated TTRTs but, all the same, they may suffer inversions from nodes of the same or lower classes.

B. Scheduling analysis for high priority (class_6) nodes

The number of nodes in class_6 shall be denoted n_6 . The period of a class_6 node must be such that, after $t_0(i)$, it can tolerate the worst case of inversions plus the transmission of messages by the upstream class_6 nodes. Inversions come from downstream class_6 nodes and from nodes belonging to lower classes receiving the token before their TTRTs have expired.

Therefore

$$\forall i = 1, 2, \dots, n_6 \quad T_i \geq n_6 + \sum_{h=n_6+1}^m f(h,i)$$

where

$$f(h,i) = \begin{cases} 1, & \text{if } R(h) > n_6 - i + \sum_{x=n_6+1}^{h-1} f(x,i) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

C. Scheduling analysis for lower classes nodes

Since the priority of a lower class node depends only on its placement and on its $TTRT$, it is irrelevant to which of the lower classes it belongs. In every case, its period must be such that, after $t_0(i)$, it can tolerate transmissions by upstream nodes with higher priorities as well as transmissions by downstream nodes producing inversions. C_i and $|C_i|$ will symbolize the set of all i downstream nodes with the same $TTRT$ as node i and the cardinality of the set, respectively. Obviously, $|C_i| = k \cdot R(i) + |C_i| \bmod R(i)$, $k \in \{0, 1, 2, \dots\}$. $|C_i| \bmod R(i)$ will be notated C_i^m . In what follows it will be assumed that in each case there are enough downstream nodes to produce the upper bound of token inversions.

Lemma 1. In the round starting at $t_0(i)$, up to $R(i)$ nodes may transmit inhibiting the following nodes.

Lemma 2. Following $t_0(i)$, all nodes $\in C_i$ will transmit before node i .

The number of node i 's placement and token inversions will be notated $p(i)$ and $\tau(i)$ respectively.

Lemma 3. $p(i) = |C_i|$

Lemma 4. Following $t_0(i)$, the number of placement inversions in the last round in which nodes $\in C_i$ transmit is either $R(i)$ or C_i^m .

The round after $t_0(i)$ in which less than $R(i)$ placement inversions take place will be notated V_0 . Topologically, it starts between nodes i and $i+1$. Successive rounds will be notated V_1, V_2, \dots

Lemma 5: If $C_i^m = 0$, then in V_0 the number of token inversions is $\tilde{R}(i)$.

Lemma 6: If $C_i^m \neq 0$ and $\tilde{R}(i) > C_i^m$, then the number of token inversions in V_0 is $\tilde{R}(i) - C_i^m$.

Lemma 7: If $C_i^m \neq 0$ and $\tilde{R}(i) \leq C_i^m$, then no token inversions take place in V_0 .

Lemma 8: The number of placement inversions plus the number of token inversions in V_0 , symbolized L_0 , is $\max(\tilde{R}(i), C_i^m)$.

Lemma 9: The number of token inversions in V_0 is $L_0 - C_i^m$

By Lemma 9, the total number of inversions suffered by node i in V_0 will be less than $R(i)$ and therefore it would be able to transmit in that round. Messages, however, may be generated in the i upstream nodes in number such that, added to the inversions, inhibit node i . In the next rounds, and although they have heard high priority messages, some i downstream nodes may still be able to transmit, producing new inversions. The process goes on until no more inversions are possible. The round in which this occurs is found in the next theorems.

Theorem 1: If they exist, token inversions take place up to and including the round V_r , where $r = \max x \in \mathbb{N} \cup \{0\} \mid x(\tilde{R}(i) - R(i)) + L_0 > 0$.

Theorem 2: The total number of token inversions that node i may suffer is

$$\tau(i) = \frac{(r^2 + r)(\tilde{R}(i) - R(i))}{2} + (r + 1) \cdot \max(\tilde{R}(i), C_i^m) - C_i^m$$

Corollary: The total number of inversions (placement plus token) that node i may suffer is

$$I(i) = p(i) + \tau(i) = |C_i| + \frac{(r^2 + r)(\tilde{R}(i) - R(i))}{2} + (r + 1) \cdot \max(\tilde{R}(i), C_i^m) - C_i^m$$

V. Schedulability test

For each node belonging to the lower classes, the expression above provides information necessary to test if the network is schedulable. Each of those nodes must tolerate its own worst case of inversions plus the load coming from nodes with higher priority. Its period must then be

bigger than or equal to the $(I(i)+1)^{\text{th}}$ empty slot in the system of $(i-1)$ nodes. This necessary condition is expressed as

$$\forall i \in \{n_0 + 1, \dots, m\} \quad T_i \geq e_{(I(i)+1)(i-1)} \quad (2)$$

$$\text{where } e_{(I(i)+1)(i-1)} = \text{least } t | t = 1 + I(i) + \sum_{h=1}^{i-1} \left\lceil \frac{t}{T_h} \right\rceil$$

If expressions (1) and (2) hold, the system is RM schedulable.

VI. Conclusions

In general, any method to test the RM schedulability of a LAN needs, as necessary data, the number of priority inversions that each node may suffer. Although timed-token protocols, a taxonomic family to which 802.4 belongs, have been extensively studied, results obtained for other protocols, e.g. FDDI, are by far not directly translatable to it. The presented approach is highly theoretical but practical applications are immediate. In fact, it is very easy to program the final expressions and to carry out the test on the RM schedulability of an 8802/4 network using the empty-slots method. The importance of the result is obvious since, on one hand, 8802/4 has been adopted for the physical layer and the MAC sublayer of the Manufacturing Automation Protocol, and, on the other hand, RMS is a *de facto* RT priority standard supported, among others, by the USA Department of Defense.

References

- [1] IEEE Std. 802.4. "Token-Passing Bus Access Method and Physical Layer Specifications". *Institute of Electrical and Electronics Engineers, Inc.* 1985.
- [2] Janetzky, D. and Watson, K.S.. "Performance evaluation of the MAP Token-Bus in real-time applications". *Advances in Local Area Networks*, IEEE Press, New York, pp 411-427. 1987.

- [3] Joseph, M. and Pandya, P. "Finding response times in a real time system". *The Computer Journal*, 29,5, pp. 390-395. 1986.
- [4] Lehozcky, J. P., Sha, L. and Ding, Y. "The rate monotonic scheduling algorithm: Exact characterization and average case behavior". In *Proc. Real Time Systems Symp.*, IEEE CS, Los Alamitos, CA. 1989.
- [5] Liu, C.L. and Layland, J.W. "Scheduling algorithms for multiprogramming in hard real time environments". *J. ACM*, 20,1, pp. 46-61. 1973.
- [6] Orozco, J. and Santos, J. "Diagramación de tareas en sistemas operativos de tiempo real PMC con inversión de prioridades". In *Proc. 23 Jornadas Argentinas de Informática e Investigación Operativa*, pp. 1.25-1.35. 1994.
- [7] Rzehak, H., Elnakhal, A. and Jaeger R. "Analysis of Real-Time Properties and Rules for Setting Protocol Parameters of MAP Networks". *The Journal of Real-Time Systems*, 1,3, pp. 221-241. 1989.
- [8] Santos, J., Gastaminza, M.L., Orozco, J., Picardi, D. and Alimenti, O. "Priorities and protocols in hard real-time LANs: implementing a crisis-free system". *Computer Communications*, 14,9, pp. 507-514. 1991.
- [9] Santos, J., Gastaminza, M.L., Orozco, J. and Matrangolo, C. "802.5 priority mechanism in hard real-time RMS applications". *Computer Communications*, 17,6, pp. 439-442. 1994.