EDD Algorithm Performance Guarantee for Periodic Hard-Real-Time Scheduling in Distributed Systems

Maurizio A. Bonuccelli, M. Claudia Clò
Dipartimento di Informatica
Università di Pisa
Corso Italia 40, 56125 Pisa, Italy
Phone: +39-050-887255; Fax: +39-050-887226
bonucce@di.unipi.it, clo@di.unipi.it

Abstract

In this paper, we investigate the worst case performance of Earliest Due Date algorithm when applied to packet scheduling in distributed systems. We assume that the processing elements communicate via a multistage interconnection network, and that the system is synchronous. When two or more packets are simultaneously sent over the same input port, or received through the same output port, the packets undergo a collision and are damaged, needing to be retransmitted later. This causes a performance degradation in terms of both throughput and delay. So, collisions must be avoided. The special type of traffic to be scheduled by Earliest Due Date is a periodic hard-real time one, and the objective is to schedule all the packets within their individual due dates. We establish that EDD is always able to produce a schedule meeting this objective, whenever the so called link utilization is no more than 1/2, showing that this worst case performance bound is tight. Such a bound can be effectively used as a feasibility test before actually running the algorithm.

1. Introduction

In this paper, we consider distributed memory processing systems centered around multistage interconnection networks. Such systems are formed by a set of processing elements, each equipped with its own memory modules, and communicating by means of an interconnection network.

We assume that such a communication is packetized and synchronous, and that the network is a nonblocking permutation one. The processing elements are connected to the interconnection network by means of input and output ports.

Obviously, each input and output port cannot be involved in more than one transmission at a time. Thus, we can have concurrent transmissions, provided that they do not involve the same port more than once. When this is not the case, all the transmissions with common ports are damaged, since a $collision$ occurs. Collided transmissions are useless and must be discarded and repeated later. This is a source of performance degradation, both from a system point of view (the network is used for transmitting messages that must be discarded) and from a user perspective (the messages undergo a longer delay before being successfully received). Such a performance degradation can be very serious, and so collisions should be avoided. Collision avoidance can be done by a proper scheduling of the messages. For a given set of messages, several different schedules exist, each with a specific impact on the overall system performance.

The same kind of problem occurs in a wide variety of systems, called multichannel single-hop communication systems. Examples of these are the assumed multistage interconnection networks as well as WDM purely optical networks [1, 2, 3, 6, 10], when operated under a Time Division Multiplexing (TDM) scheme. As usual in TDM schemes, we assume that communication is organized into $frames$. Each frame is divided into equal length *time slots*, representing the transmission of unitary blocks of information, called *packets*. Packet scheduling is often called *time slot assignment* (TSA for short), and has been extensively studied in the past in a communication setting, by considering particular side constraints, introduced in the basic problem to take into account specific system organizations, like intersatellite links, or tuning time in optical networks, and special traffic features, like different priority messages (e.g., see [1, 2, 3, 6, 10, 18, 20, 7, 8, 23, 12, 19, 11]). Generally, the target is to minimize the schedule length.

Real time communication main feature is to require a timely delivery of periodic messages (each message must be periodically scheduled for transmission over an unlimited time interval) to the destination. Such a timely delivery is represented by a due date (also called deadline) associated to each message. Due date meeting is essential for ensuring a safe system working, or reaching a minimum quality level, acceptable by the end user.

The goal of the packet scheduling problem investigated in this paper, called *Periodic Hard Real Time Time Slot Assignment*, or *PHRT-TSA* for short, is to find a *feasible schedule*, whenever it exists. By feasible schedule we mean a schedule meeting the due dates of all the messages. This is the most important goal in real time settings, where high delays are not tolerated. Processor scheduling of uni- and multi-processor computing systems for real time applications has been widely studied since the early 70's (a survey can be found in [13, 14, 15, 16, 21]), mainly under the objective function considered here.

In spite of its rapidly growing importance, real time packet scheduling has received some attention only very recently. An earlier attempt to study the TSA problem of multimedia traffic can be found in [6]. Real time transmissions with explicit deadlines have been investigated [23]. In that paper, however, a model different from the one assumed here is considered, since message release times are unknown. In [17], a slightly more general scheduling problem, called *Multiple-Period TSA* (*MP-TSA*), is considered. The model allows two or more messages with different periods to be transmitted from the same origin to the same destination. Four efficient heuristic algorithms are presented and the success rate of each algorithm is evaluated through simulation.

In [9] other two heuristic algorithms are proposed for the MP-TSA problem, *Nested Period Scheduling* (*NPS*) and *Slot-by-Slot Earliest-Deadline-First Earliest-Arrival-First* (*SS-EDF-EAF*, EDF is a synonym of EDD). Like in [17], multiple messages with different periods are allowed to be transmitted, and each message stream is formed by only one packet. So, such message streams are a little different from those considered in this paper (We assume single stream, multiple packet messages). In [9], it is shown that, if each period evenly divides all larger periods and if the maximum link utilization is not larger than one, the NPS algorithm can always find a feasible schedule. Notice that the NPS algorithm has a pseudopolynomial computational complexity. Besides, it is shown that NPS and SS-EDF-EAF can schedule traffic with arbitrary periods if the link utilization is not larger than $1/4$ and $1/14$, respectively.

Finally, in [4], the non periodic version of the problem considered in this paper, has been investigated. It is shown that the problem is computationally intractable (NP-complete) even in very restricted cases. Heuristics are proposed, and their performance (in terms of utilization factors) is presented.

In this paper, we consider the well known Earliest Due Date (EDD) heuristic, also called Earliest Deadline First, EDF, like in [9], and assess its worst case performance. This heuristic is based on the same policy widely used in different (e.g. shared memory multiprocessor) settings [22]. In Section 2, we formally state the problem. Besides, we give some conditions, necessary for producing a feasible schedule, and present a counterexample to a conjecture formulated by other authors. Then, we propose EDD (Section 3), and establish a tight worst case performance bound (in terms of utilization factors) for it. Performance guarantees are very important in this setting, since fast tests assessing whether the algorithm will produce a schedule meeting all the deadlines or not, without actually running the algorithm (a time consuming process), can be derived from them. This gives insights in the system design stage, also. Finally, conclusions and open problems terminate the paper.

2. PHRT-TSA Problem

We now present the mathematical formulation of the PHRT-TSA problem for single-hop networks. Messages are periodic and each has an associated deadline. Each message needs to be transmitted in each period within its deadline, in order to guarantee the system safety. Besides, when two or more messages are transmitted simultaneously over the same channel, or from the same origin or to the same destination, a message collision occurs. To avoid message collisions, a proper schedule of the messages must be devised.

An $n \times n$ nonnegative integer *traffic matrix* $T = (t_{ij})$ models the traffic between each origin and destination, $1 \leq i, j \leq n$. Rows and columns of matrix T represent system origins and destinations, respectively.
An $n \times n$ nonnegative integer *deadline matrix* $D = (d_{ij})$ contains all the deadlines corresponding to the matrix T entries, $1 \leq i, j \leq n$, where d_{ij} corresponds to t_{ij}. Let r_{ij} be the release time of entry t_{ij}. The traffic is *periodic*. A period p_{ij} is associated to each message t_{ij}. A message has to be retransmitted once in each period. Hence, the first and k-th, $k > 1$, occurrences of a periodic message t_{ij} take place in time intervals $[r_{ij}, d_{ij}]$ and $[r_{ij} + (k-1)p_{ij}, d_{ij} + (k-1)p_{ij}]$, respectively. Obviously, the deadline, release time and period of message t_{ij} are related in the following way:

$$\begin{aligned} r_{ij} &< d_{ij} \leq r_{ij} + p_{ij} \\ t_{ij} &\leq d_{ij} \leq p_{ij} \end{aligned}$$

In this paper, we shall assume that $r_{ij} = 0$ and $p_{ij} = d_{ij}$. Hence, a message t_{ij} must be scheduled for transmission in each interval $[(k-1)d_{ij}, kd_{ij}]$, $k \geq 1$. This simplifying assumption is commonly made in real time scheduling [13, 14, 15, 16, 21].
In order to meet the traffic timing requirements, we have the following *constraint* on the schedule:

C1: a message transmission has neither to start before its release time, nor to be completed after its deadline, during each period.

Note that a deadline kd_{ij} is both the deadline of the k-th message occurrence and the release time of $(k+1)$-th one.

In order to avoid message collisions, traffic transmission must meet the following *constraint*:

C2: in each time slot, at most one message can be scheduled for transmission over one channel, or from an origin or to a destination.

Given the matrices T and D, let us consider the composition $\frac{T}{D} = \left(\frac{t_{ij}}{d_{ij}}\right)$. A *feasible schedule* for the PHRT-TSA problem can be defined in the following way:

Definition 1 Consider the composition $\frac{T}{D}$; after each time slot scheduling, matrices T and D are modified in the following way. Let $t_{ij}^{(0)}, t_{ij}^{(a)}$ and $t_{ij}^{(a+1)}$ be the values of the entry t_{ij} at the initial time $t = 0$, at an arbitrary time a and at time $a + 1$. Similarly, let $d_{ij}^{(0)}, d_{ij}^{(a)}$ and $d_{ij}^{(a+1)}$ be the values of the corresponding entry of D at the same times. Then:

$$t_{ij}^{(a+1)} = \begin{cases} t_{ij}^{(a)} - 1 & \text{if } t_{ij} \text{ is scheduled at time } a \\ t_{ij}^{(a)} & \text{if } t_{ij} \text{ is \underline{not} scheduled at time } a \\ t_{ij}^{(0)} & \text{if time } a+1 \text{ is a multiple of } d_{ij} \end{cases}$$

$$d_{ij}^{(a+1)} = \begin{cases} d_{ij}^{(a)} - 1 & \text{if } t_{ij}^{(a+1)} > 0 \\ 0 & \text{if } t_{ij}^{(a+1)} = 0 \\ d_{ij}^{(0)} & \text{if time } a+1 \text{ is a multiple of } d_{ij} \end{cases}$$

In this case, a feasible solution of the PHRT-TSA problem is a decomposition of $\frac{T}{D}$, i.e. a sequence of matrices $\frac{T}{D}$, $\frac{T'}{D'}$, $\frac{T''}{D''}$, $\cdots$, meeting the above constraints C1 and C2.

Hence, the **PHRT-TSA** problem can be defined in the following way:

INSTANCE: an $n \times n$ traffic matrix T, an $n \times n$ deadline matrix D

SOLUTION: a feasible schedule

Let Δ be the least common multiple of the deadlines: $\Delta = \text{l.c.m.}\{d_{ij} : 1 \leq i, j \leq n\}$.

<u>Remark</u>

Because of the traffic periodicity, we should address the TSA problem indefinitely. So, a feasible schedule is an *infinite sequence* of matrices. However, it is possible to reduce the problem over a finite time interval. Indeed, after an interval with length equal to Δ, the entries are all again ready for transmission, like at the initial time $t = 0$. So it sufficed to find a feasible schedule for the finite time interval $[0, \Delta]$.

Appendix A shows a feasible schedule for matrices T and D.

Let us now give some definitions useful in establishing some properties of the PHRT-TSA problem.

Definition 2 A *switching matrix* S is a traffic matrix with at most one nonzero entry in each line.

Definition 3 Let r_i and c_j denote the *row* and *column sums* of $\frac{T}{D}$, respectively ($1 \leq i, j \leq n$):

$$r_i = \sum_{h=1}^{n} \frac{t_{ih}}{d_{ih}} \ , \ c_j = \sum_{h=1}^{n} \frac{t_{hj}}{d_{hj}}$$

Definition 4 A matrix line is called *exposed* if all entries in that line are zero and *covered* if there is at least one non-zero entry in it.

Definition 5 A *doubly stochastic matrix* $S = (s_{ij})$ is a nonnegative integer matrix with $r_i = c_j = 1$, $\forall i$, $\forall j$.

Definition 6 The *maximum line sum* of matrix $\frac{T}{D}$ is defined as the maximum row or column sums: $l_{\max} = \max_{1 \leq i,j \leq n}\{r_i, c_j\}$

We now give a definition of the PHRT-TSA problem for the special case in which there exist only two different kind of deadlines: $d_{ij} \in \{d_1, d_2\}$, $\forall (i, j)$.

Definition 7 Let $X_a(i)$ and $Y_a(j)$ be the *sum of the traffic entries in row* i and *in column* j with deadline d_a ($a \in \{1, 2\}$):

$$X_a(i) = \sum_{1 \leq h \leq n : d_{ih} = d_a} t_{ih} \ , \ Y_a(j) = \sum_{1 \leq h \leq n : d_{hj} = d_a} t_{hj}$$

Clearly, the following equalities hold:

$$r_i = \frac{X_1(i)}{d_1} + \frac{X_2(i)}{d_2} \ , \ c_j = \frac{Y_1(j)}{d_1} + \frac{Y_2(j)}{d_2}$$

2.1. Necessary Condition

Let us consider an instance $\frac{T}{D}$ of the PHRT-TSA problem. An obvious necessary condition for the schedulability is the following $(1 \leq i, j \leq n)$:

$$\frac{t_{ij}}{d_{ij}} \leq 1 \tag{1}$$

We now give a less obvious necessary condition for the schedulability of $\frac{T}{D}$.

If a feasible schedule for the PHRT-TSA problem exists, then the following conditions must hold (see [5]):

$$\left\{ \begin{array}{l} \sum_{j=1}^{n} \frac{t_{ij}}{d_{ij}} \leq 1 \ \ (1 \leq i \leq n) \\ \sum_{i=1}^{n} \frac{t_{ij}}{d_{ij}} \leq 1 \ \ (1 \leq j \leq n) \end{array} \right. \tag{2}$$

These inequalities can be rewritten in terms of row and column sums, r_i and c_j:

$$\left\{ \begin{array}{lll} r_i & \leq & 1 \ \ (1 \leq i \leq n) \\ c_j & \leq & 1 \ \ (1 \leq j \leq n) \end{array} \right. \tag{3}$$

If conditions (2) (or (3)) hold with the $<$ sign for each i and j, we say that the *strict conditions* are met. On the contrary, if all the row and column sums are equal to 1, then $\frac{T}{D}$ is a *doubly stochastic* matrix.

Remark

The conjecture that all the matrices meeting condition (2) always have a feasible schedule has been made [17, 9]. No proof neither counterexample to this conjecture is known. The conjecture means that the necessary condition (2) (or (3)) for the schedulability is also a sufficient one. Here, we show a *counterexample*, i.e. matrices satisfying the necessary condition, which are not schedulable, thus disproving it.

Let us consider the following matrix $\frac{T}{D}$:

$$\left[\begin{array}{cccc} 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 1/2 & 1/6 & 0 \\ 1/3 & 0 & 1/2 & 1/6 \\ 1/3 & 1/6 & 0 & 1/2 \end{array} \right] \tag{4}$$

It meets the necessary condition (2), but no feasible schedule exists for it (see Appendix B). So, the conjecture is not true, i.e. the necessary condition is not a sufficient one.

3. Performance Guarantees

We can now prove a feasibility condition for algorithm *Earliest Due Date* (*EDD*).

First, we establish a result to be used later.

Let us assume to have only two deadline values, d_1 and d_2, and let $d_1 < d_2$. Consider the k-th occurrence of period d_2, i.e. interval $[(k-1)d_2, kd_2]$. Denote by $I_1, I_2, \cdots, I_q$ the subintervals between two deadlines in that interval, i.e. $I_1 = [(k-1)d_2, hd_1]$, $I_2 = [hd_1, (h+1)d_1], \cdots$, $I_q = [fd_1, kd_2]$. Let $|I_i|$ denote the length of interval I_i. Note that $|I_i| = d_1$ for $2 \leq i \leq q-1$, $|I_1| < d_1$, $|I_q| < d_1$. Let $m = \lfloor \frac{d_2}{d_1} \rfloor$. Then $d_2 = md_1 + r$ $(r < d_1)$.

Lemma 1 *Number N of subintervals of length equal to d_1, i.e. subintervals $I_2, \cdots, I_{q-1}$, in $[(k-1)d_2, kd_2]$, is*

$$N = \left\{ \begin{array}{ll} m & \textit{if } |I_1| + |I_q| < d_1 \\ m-1 & \textit{if } |I_1| + |I_q| \geq d_1. \end{array} \right.$$

Besides, if $|I_1| + |I_q| < d_1$, then $|I_1| + |I_q| = r$, otherwise $|I_1| + |I_q| = d_1 + r$.

The proof of this lemma can be found in [5].

Example 1 Let $d_1 = 5, d_2 = 17$. Then $m = 3$ and $r = 2$. Consider interval $[34, 51]$: $I_1 = [34, 35]$ and I_q=[50,51]. So $|I_1| + |I_q| = r < d_1$, and $N = m = 3$. In fact: I_2 =[35,40], I_3= [40,45] and I_4= [45,50].

Consider interval $[17, 34]$. In this case $I_1 = [17, 20]$ and I_q=[30,34]. Hence $|I_1| + |I_q| > d_1$. Note that $|I_1| + |I_q| = d_1 + r$. Then, $N = m - 1 = 2$, i.e. I_2 =[20,25] and I_3= [25,30].

3.1. Algorithm EDD

Algorithm *Earliest Due Date* (*EDD*) for our periodic traffic problem schedules the entries t_{ij} $(t_{ij} > 0)$ according to the values of the next deadline, earliest first, meeting the switching matrix constraints.

The algorithm is a dynamic-priority scheduling one, so the entry priorities are dynamically updated, after each time slot assignment. A detailed algorithm description is given in Appendix C.

The initial values of traffic and deadline matrices are maintained in matrices T and D, which are not modified by the algorithm. EDD uses a matrix $T_1 = (t^1_{ij})$ for generating the switching matrix $S_k = (s^k_{ij})$. Two support matrices, $T_2 = (t^2_{ij})$ and $D_2 = (d^2_{ij})$ are used for maintaining the current values of traffic and deadline entries.

Procedure *Sort* is applied to deadline matrix D_2 and outputs a sorted list of its entries. The priorities are assigned to entries t_{ij} on the basis of this sorted list. Since EDD is a dynamic-priority algorithm, the sorting of D_2 is performed each time a switching matrix is built by the algorithm. Procedure *Sort* complexity is O($n^2 \log n$). O(n^2) time is required to scan the $list$, and O(n^3) in total to generate a switching matrix. If we restrict over a finite time interval, namely we generate the schedule only in $[0, \Delta]$, the number of switching matrices depends on the number of deadline occurrences until Δ.

We now establish the following feasibility condition for algorithm EDD when $d_{ij} \in \{d_1, d_2\}$, $d_1 < d_2$.

Theorem 1 *Let a matrix* $\frac{T}{D} = \left(\frac{t_{ij}}{d_{ij}}\right)$ *be given, with* $d_{ij} \in \{d_1, d_2\}$*. If*

$$r_i \leq \frac{1}{2}\ ,\ c_j \leq \frac{1}{2}$$

for $1 \leq i, j \leq n$*, then EDD always generates a feasible schedule for the PHRT-TSA problem.*

Proof: Assume that $r_i \leq \frac{1}{2}$, $\forall i$, and $c_j \leq \frac{1}{2}$, $\forall j$. Consider the schedule generated by EDD. Ab absurdo, assume that some deadlines are not met in that schedule, and let $\delta = hd_{ij}$ be the first of such deadlines ($h \geq 1$). That is, the h-th occurrence of entry t_{ij} is not entirely transmitted in $[(h-1)d_{ij}, hd_{ij}]$, while all the previous occurrences of t_{ij} and of all the other entries meet their deadlines. We want to show that this assumption leads to a contradiction. We have two cases according to $d_{ij} = d_1$ or $d_{ij} = d_2$.

Case 1) Assume that $d_{ij} = d_2$, i.e. $\delta = hd_2$ ($h \geq 1$). We are considering row i and column j: so, for the sake of simplicity, we omit the indexes i and j in $X_a(i)$ and $Y_a(j)$.

If there are switching matrices in the schedule between 0 and hd_2 with row i and column j both exposed, then let $\bar{s}$ be the last of such matrices and assume that $\bar{s} \in [(k-1)d_2, kd_2]$, $0 \leq k < h$. We consider time interval $[kd_2, hd_2]$. If either row i or column j are always covered until time hd_2, then $k = 0$.
X_2 and Y_2 must be entirely scheduled in each period of length d_2. The longest time taken to schedule type 2 entries is when X_2 and Y_2 are never transmitted in parallel, i.e. a type 2 entry (namely an entry with deadline d_2) in row i and an entry of type 2 in column j are never transmitted in the same time slot. So, $X_2 + Y_2$ represents the longest time taken to transmit all the type 2 entries in row i and column j during each period of length d_2. Hence, in $[kd_2, hd_2]$, the time taken to schedule type 2 entries is at least $(h-k)(X_2 + Y_2)$.
Let $f = \lfloor \frac{(h-k)d_2}{d_1} \rfloor$, and let $I_1, I_2, \cdots, I_q$ be all the subintervals in $[kd_2, hd_2]$ between any two consecutive d_1 deadlines. From Lemma 1, it results that the number of such subintervals is:

$$\begin{array}{ll} q = f & \text{if } |I_1| + |I_q| = r \\ q = f - 1 & \text{if } |I_1| + |I_q| = d_1 + r \end{array}$$

In the second case, when $|I_1| + |I_q| = d_1 + r$, we can assume that type 1 entries (entries with deadline d_1) are scheduled at least one more time in I_1 and I_q since $|I_1| + |I_q| > d_1$. In both cases, f is a lower bound on the number of times that type 1 entries must be scheduled in $[kd_2, hd_2]$.

For type 1 entries, we can repeat the argument made above for type 2 entries. So, in $[kd_2, hd_2]$ at most $X_1 + Y_1$ slots are devoted to the transmission of type 1 entries for each of the f subintervals above.

Then $(h-k)(X_2 + Y_2) + f(X_1 + Y_1)$ is a lower bound on the entries that must be scheduled for transmission during interval $[kd_2, hd_2]$, in order to meet all the deadlines. Since t_{ij} is not entirely transmitted within $[(h-1)d_2, hd_2]$, it results: $(h-k)(X_2 + Y_2) + f(X_1 + Y_1) > (h-k)d_2$, and

$$\frac{(h-k)(X_2 + Y_2)}{(h-k)d_2} + \frac{f(X_1 + Y_1)}{(h-k)d_2} > 1 \tag{5}$$

This can be rewritten in the following way:

$$\frac{X_2}{d_2} + \frac{Y_2}{d_2} + \frac{fX_1}{(h-k)d_2} + \frac{fY_1}{(h-k)d_2} > 1$$

One of the following two inequalities must hold:

$$\frac{fX_1}{(h-k)d_2} + \frac{X_2}{d_2} > \frac{1}{2} \tag{6}$$

or

$$\frac{fY_1}{(h-k)d_2} + \frac{Y_2}{d_2} > \frac{1}{2} \tag{7}$$

Note that, since $f = \lfloor \frac{(h-k)d_2}{d_1} \rfloor \leq \frac{(h-k)d_2}{d_1}$, then $\frac{f}{(h-k)d_2} \leq \frac{1}{d_1}$. So, by putting it in inequality (6), we obtain:

$$\frac{X_1}{d_1} + \frac{X_2}{d_2} \geq \frac{fX_1}{(h-k)d_2} + \frac{X_2}{d_2} > \frac{1}{2}$$

Hence $r_i > \frac{1}{2}$.

Similarly, from inequality (7), we obtain $c_j > \frac{1}{2}$. Therefore, one of the following inequalities must hold: $r_i > \frac{1}{2}$ or $c_j > \frac{1}{2}$, which is a contradiction.

Case 2) Assume that $\delta = fd_1$ ($f \geq 1$). Consider interval $[0, fd_1]$. If i and j are both exposed in a switching matrix $\bar{s}$, then each entry in row i and column j has been scheduled before $\bar{s}$ and the next deadlines of the entries in those lines are met. If $\bar{s} \in [(a-1)d_1, ad_1]$, then consider interval $[ad_1, fd_1]$, $0 \leq a < f$. If either row i or column j are always covered until time fd_1, then $a = 0$. X_1 and Y_1 must be entirely scheduled in each interval of length d_1. The longest time taken to schedule type 1 entries occurs when X_1 and Y_1 are never transmitted in parallel. Hence, $X_1 + Y_1$ represents the longest time taken to transmit all the entries of type 1 in row i and column j during each interval as above. In $[ad_1, fd_1]$ at least $X_1 + Y_1$ entries should be scheduled $(f-a)$ times. We distinguish two cases:

2.1) $(f-a)\, d_1 \leq d_2$
2.2) $(f-a)\, d_1 > d_2$

2.1) Let us assume that $(f-a)\, d_1 \leq d_2$, and assume that a deadline d_2 (say kd_2) occurs in $[ad_1, fd_1]$. Then kd_2 is the next occurrence of a d_2 deadline after $\bar{s}$, in which row i and column j are both exposed. Hence, all type 2 entries have already been scheduled before $\bar{s}$: in time interval $[ad_1, kd_2]$

there are no type 2 entries to be scheduled for transmission. Since $(f-a)d_1 \leq d_2$, no other d_2 deadline occurs in time interval $[ad_1, fd_1]$.

Algorithm EDD gives higher priority to the entries with earlier deadline. So, in $[kd_2, fd_1]$ it gives higher priority to type 1 entries over type 2 ones. Entry t_{ij} is not entirely scheduled in $[(f-1)d_1, fd_1]$, and this is due to the scheduling of other type 1 entries in its row or column. Then, we do not consider the scheduling of type 2 entries, and a lower bound on the entries scheduled in $[ad_1, fd_1]$ is given by $(f-a)(X_1+Y_1)$. So $(f-a)(X_1+Y_1) > (f-a)d_1$ and $\frac{X_1+Y_1}{d_1} > 1$. Hence we have: $\frac{X_1}{d_1} > \frac{1}{2}$ or $\frac{Y_1}{d_1} > \frac{1}{2}$. So $r_i > \frac{1}{2}$ or $c_j > \frac{1}{2}$, which is a contradiction. A similar argument holds when no type d_2 deadline is within $[ad_1, fd_1]$

2.2) Assume now that $(f-a)d_1 > d_2$.

In this case, interval $[ad_1, fd_1]$ contains one or more d_2 deadlines.

Let $h = \lfloor \frac{(f-a)d_1}{d_2} \rfloor$. For type 2 entries, we can repeat the argument used for type 1 entries in case 1. Then, in $[ad_1, fd_1]$, at least $h(X_2+Y_2)$ slots are devoted to the transmission of type 2 entries. Since deadline fd_{ij} is not met, then $(f-a)(X_1+Y_1)+h(X_2+Y_2) > (f-a)d_1$, that is $\frac{X_1}{d_1}+\frac{Y_1}{d_1}+\frac{hX_2}{(f-a)d_1}+\frac{hY_2}{(f-a)d_1} > 1$. So, one of the following two inequalities must hold:

$$\frac{X_1}{d_1}+\frac{hX_2}{(f-a)d_1} > \frac{1}{2} \tag{8}$$

or

$$\frac{Y_1}{d_1}+\frac{hY_2}{(f-a)d_1} > \frac{1}{2} \tag{9}$$

Note that $h = \lfloor \frac{(f-a)d_1}{d_2} \rfloor \leq \frac{(f-a)d_1}{d_2} \Rightarrow \frac{h}{(f-a)d_1} \leq \frac{1}{d_2}$. Then, by putting it in (8), we obtain: $\frac{X_1}{d_1}+\frac{X_2}{d_2} \geq \frac{X_1}{d_1}+\frac{hX_2}{(f-a)d_1} > \frac{1}{2}$. So $r_i > \frac{1}{2}$.

Similarly, from inequality (9), we have $c_j > \frac{1}{2}$. Then, either $r_i > \frac{1}{2}$ or $c_j > \frac{1}{2}$, which is a contradiction. It follows that EDD always meets all deadlines and the schedule is feasible. QED

Note that Theorem 1 states a sufficient but not necessary condition for the schedulability of algorithm EDD.

In order to show that $1/2$ is *asymptotically* achievable, let us consider the following $n \times n$ $\frac{T}{D}$ matrix, where $d_1 = 2$ and $d_2 = x$ $(x > 2)$:

$$\begin{bmatrix} 1/x & 1/x & 1/x & 1/x & \cdots \\ 1/2 & - & - & - & - \\ - & 1/2 & - & - & - \\ - & - & 1/2 & - & - \\ - & - & - & 1/2 & - \\ - & - & - & - & \cdots \end{bmatrix} \tag{10}$$

Lemma 2 *Algorithm EDD generates a feasible schedule for matrix (10) if and only if*

$$d_2 \begin{cases} \geq 2n-2 & \text{if } d_2 \text{ is even} \\ \geq 2n-3 & \text{if } d_2 \text{ is odd} \end{cases}$$

where n is the matrix size.

The proof of this lemma can be found in [5]. The lemma states that if $d_2 < 2n-3$, then EDD does not generate a feasible schedule for the matrix. We want to show that performance bound $1/2$ is asymptotically achievable. So, let us assume that $d_2 = 2n-4$. Thus, the maximum line sum for this matrix is

$$l_{\max} = \max\left\{ \frac{n-1}{d_2} \, , \, \frac{1}{d_2}+\frac{1}{2} \right\}$$

From lemma 2, if $d_2 = 2n-4$, then EDD fails:

$$\frac{n-1}{d_2} < \frac{1}{d_2}+\frac{1}{2} = \frac{1}{2(n-2)}+\frac{1}{2} = \frac{n-1}{2(n-2)} \stackrel{n\to\infty}{\longrightarrow} \frac{1}{2}$$

Theorem 1 can be extended to the case with three deadline values.

Theorem 2 *Let a matrix* $\frac{T}{D} = \left(\frac{t_{ij}}{d_{ij}}\right)$ *be given, with* $d_{ij} \in \{d_1, d_2, d_3\}$. *If*

$$r_i \leq \frac{1}{2} \, , \; c_j \leq \frac{1}{2}$$

for $1 \leq i, j \leq n$ *, then EDD always produces a feasible schedule for the PHRT-TSA problem.*

Proof: We give only a sketch of the proof, since it is very similar to that of two deadlines case. Assume that there are three deadline values, d_1, d_2, d_3, and that:

$$d_1 < d_2 < d_3.$$

Ab absurdo, assume that some deadlines are not met in the schedule generated by EDD, and let $\delta = hd_{ij}$ be the first of such deadlines. We consider time interval $[0, hd_{ij}]$ if row i or column j are always covered in the interval. If there are switching matrices in which row i and column j are both exposed, then let $\bar{s}$ be the last of such matrices and assume that $\bar{s} \in [(a-1)d_{ij}, ad_{ij}]$, $a < h$. Then, the deadlines of all the entries in row i or column j occurring before $\bar{s}$ are met, and we consider interval $[ad_{ij}, hd_{ij}]$.

On the basis of the deadline value, we have the following cases:

Case 1) $\delta = hd_3$

Case 2) $\delta = hd_2$

2.1) $(h-a)d_2 < d_3$

2.2) $(h-a)d_2 \geq d_3$

Case 3) $\delta = hd_1$
3.1) $(h-a)d_1 < d_2$
3.2) $d_2 \le (h-a)d_1 < d_3$
3.3) $(h-a)d_1 \ge d_3$

We present only case 3 proof since the proofs of the other cases are very similar to those of Theorem 1.

case 3) Assume that $\delta = hd_1$. Again, consider $[ad_1, hd_1]$. Then, $(h-a)(X_1+Y_1)$ represents the type 1 entries that must be scheduled in $[ad_1, hd_1]$. We consider the subcases 3.1, 3.2 and 3.3.

3.1) Assume that $(h-a)d_1 < d_2$. Then we also have: $(h-a)d_1 < d_3$. Assume that kd_3 and fd_2 occur in $[ad_1, hd_1]$: they are the earliest type 3 and type 2 deadlines after $\bar{s}$. Hence, all type 3 and type 2 deadlines are met before $\bar{s}$. Since $(h-a)d_1 < d_2 < d_3$, no other d_2 and d_3 deadlines are in $[ad_1, hd_1]$. EDD gives higher priority to type 1 entries over type 2 and type 3 ones in $[ad_1, hd_1]$. So, we do not consider the scheduling of type 2 and type 3 entries in $[ad_1, hd_1]$. Since hd_1 is not met, it results $(h-a)(X_1+Y_1) > (h-a)d_1$, so $\frac{X_1}{d_1} > \frac{1}{2}$ or $\frac{Y_1}{d_1} > \frac{1}{2}$. That is $r_i > \frac{1}{2}$ or $c_j > \frac{1}{2}$, which is a contradiction.

3.2) Assume that $d_2 \le (h-a)d_1 < d_3$. Let $l_{12} = \lfloor\frac{(h-a)d_1}{d_2}\rfloor$. We can repeat the above argument, by considering also type 2 entries, which must be scheduled at least l_{12} times. $(h-a)(X_1+Y_1) + l_{12}(X_2+Y_2) > (h-a)d_1$. Since $l_{12} \le \frac{(h-a)d_1}{d_2}$, we have the following conclusion:

$$\frac{X_1}{d_1} + \frac{X_2}{d_2} \ge \frac{X_1}{d_1} + \frac{l_{12}X_2}{(h-a)d_1} > \frac{1}{2}$$

or

$$\frac{Y_1}{d_1} + \frac{Y_2}{d_2} \ge \frac{Y_1}{d_1} + \frac{l_{12}Y_2}{(h-a)d_1} > \frac{1}{2}$$

That is $r_i > \frac{1}{2}$ or $c_j > \frac{1}{2}$, which is a contradiction.

3.3) Assume that $(h-a)d_1 \ge d_3 > d_2$. Let $l_{13} = \lfloor\frac{(h-a)d_1}{d_3}\rfloor$. Then

$$(h-a)(X_1+Y_1) + l_{12}(X_2+Y_2) + l_{13}(X_3+Y_3) > (h-a)d_1$$

Hence $\frac{X_1}{d_1} + \frac{X_2}{d_2} + \frac{X_3}{d_3} > \frac{1}{2}$ or $\frac{Y_1}{d_1} + \frac{Y_2}{d_2} + \frac{Y_3}{d_3} > \frac{1}{2}$ which is a contradiction. QED

It is easy to see that Theorem 2 can be further extended to the case with an arbitrary number of deadlines:

Theorem 3 *Let a matrix* $\frac{T}{D} = \left(\frac{t_{ij}}{d_{ij}}\right)$ *be given, with* $d_{ij} \in \{d_1, d_2, \cdots, d_m\}$. *If*

$$r_i \le \frac{1}{2}\,,\ c_j \le \frac{1}{2}$$

for $1 \le i, j \le n$, *then EDD always generates a feasible schedule for the PHRT-TSA problem.*

The proof of this Theorem follows the pattern of that of Theorems 1 and 2, and can be found in [5].

4. Conclusions

In this paper, we have given a performance guarantee for EDD algorithm when applied to periodic hard real time packet scheduling in multichannel single-hop communication systems. We are currently investigating the performance guarantees of other scheduling algorithms, like Rate Monotonic and Minimum Laxity First. Since this is one of the first papers addressing this kind of problems, many questions remain open. Among the others, two very important open questions are the on-demand scheduling, namely the scheduling of requests dynamically coming, as soon as they are issued, and the scheduling of both periodic and aperiodic packets.

Acknowledgement

We thanks Riccardo Bettati for urging us to work on the hard real time setting, and for the helpful discussions.

5. Appendix

Appendix A:

Consider the following traffic and deadline matrices:

$$T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 2 \\ 3 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 2 & 2 & 0 \\ 0 & 3 & 3 \\ 6 & 0 & 6 \end{bmatrix}$$

According to the definition, we can consider the composition $\frac{T}{D}$:

$$\frac{T}{D} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 1/3 & 2/3 \\ 3/6 & 0 & 1/6 \end{bmatrix}$$

In this case, $\Delta = 6$ and we have to find a feasible schedule over time interval $[0, 6]$. If we schedule the boxed entries in time slot 1:

$$\begin{bmatrix} 1/2 & \boxed{1/2} & 0 \\ 0 & 1/3 & \boxed{2/3} \\ \boxed{3/6} & 0 & 1/6 \end{bmatrix}$$

then matrix $\frac{T}{D}$ becomes:

$$\frac{T'}{D'} = \begin{bmatrix} 1/1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 2/5 & 0 & 1/5 \end{bmatrix}$$

A complete and feasible schedule of the given matrices $\frac{T}{D}$ is the following:

$$\begin{bmatrix} 1/2 & \boxed{1/2} & 0 \\ 0 & 1/3 & \boxed{2/3} \\ \boxed{3/6} & 0 & 1/6 \end{bmatrix}_{t=0} \qquad \begin{bmatrix} \boxed{1/1} & 0 & 0 \\ 0 & \boxed{1/2} & 1/2 \\ 2/5 & 0 & \boxed{1/5} \end{bmatrix}_{t=1}$$

$$\begin{bmatrix} \boxed{1/2} & 1/2 & 0 \\ 0 & 0 & \boxed{1/1} \\ 2/4 & 0 & 0 \end{bmatrix}_{t=2} \qquad \begin{bmatrix} 0 & \boxed{1/1} & 0 \\ 0 & 1/3 & \boxed{2/3} \\ \boxed{2/3} & 0 & 0 \end{bmatrix}_{t=3}$$

$$\begin{bmatrix} 1/2 & \boxed{1/2} & 0 \\ 0 & 1/2 & \boxed{1/2} \\ \boxed{1/2} & 0 & 0 \end{bmatrix}_{t=4} \qquad \begin{bmatrix} \boxed{1/1} & 0 & 0 \\ 0 & \boxed{1/1} & 0 \\ 0 & 0 & 0 \end{bmatrix}_{t=5}$$

Appendix B:

We now show that no feasible schedule exists for matrix (4). Let us give some preliminary definitions.

Definition 8 A switching matrix is said to be *maximum* if it contains exactly one entry in each line. A switching matrix is said to be *maximal* if it has at most one entry in each line, but it is not possible to extend it.

Lemma 3 Consider a doubly stochastic matrix $\frac{T}{D}$. A feasible schedule for such matrix must contain only maximum switching matrices, corresponding to maximum allocations of the slots.

From this Lemma, we can derive an easy test for doubly stochastic matrices: if a schedule contains *a not maximum switching matrix*, then the schedule is *not feasible*.

We want to prove that there does not exist a feasible schedule for matrix (4):

$$\begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 1/2 & 1/6 & 0 \\ 1/3 & 0 & 1/2 & 1/6 \\ 1/3 & 1/6 & 0 & 1/2 \end{bmatrix}$$

Since $r_i = c_j = 1, 1 \leq i, j \leq n$, we consider schedules formed only by maximum switching matrices.

Because of the matrix symmetry, there are only three different maximum switching matrices:

$$1)\ \begin{bmatrix} 0 & \boxed{1/3} & 1/3 & 1/3 \\ \boxed{1/3} & 1/2 & 1/6 & 0 \\ 1/3 & 0 & \boxed{1/2} & 1/6 \\ 1/3 & 1/6 & 0 & \boxed{1/2} \end{bmatrix}_{t=0}$$

$$2)\ \begin{bmatrix} 0 & \boxed{1/3} & 1/3 & 1/3 \\ 1/3 & 1/2 & \boxed{1/6} & 0 \\ \boxed{1/3} & 0 & 1/2 & 1/6 \\ 1/3 & 1/6 & 0 & \boxed{1/2} \end{bmatrix}_{t=0}$$

$$3)\ \begin{bmatrix} 0 & \boxed{1/3} & 1/3 & 1/3 \\ 1/3 & 1/2 & \boxed{1/6} & 0 \\ 1/3 & 0 & 1/2 & \boxed{1/6} \\ \boxed{1/3} & 1/6 & 0 & 1/2 \end{bmatrix}_{t=0}$$

Consider matrix 1: since we look for maximum switching matrices, the next switching matrices are the following:

$$\begin{bmatrix} 0 & 0 & \boxed{1/2} & 1/2 \\ 0 & \boxed{1/1} & 1/5 & 0 \\ 1/2 & 0 & 0 & \boxed{1/5} \\ \boxed{1/2} & 1/5 & 0 & 0 \end{bmatrix}_{t=1}$$

$$\begin{bmatrix} 0 & 0 & 0 & \boxed{1/1} \\ 0 & 1/2 & \boxed{1/4} & 0 \\ \boxed{1/1} & 0 & 1/2 & 0 \\ 0 & \boxed{1/4} & 0 & 1/2 \end{bmatrix}_{t=2}$$

$$\begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 1/1 & 0 & 0 \\ 1/3 & 0 & 1/1 & 0 \\ 1/3 & 0 & 0 & 1/1 \end{bmatrix}_{t=3}$$

and the resulting schedule is not feasible since the last matrix cannot be scheduled without missing a deadline.

Consider now matrix 2: the following switching matrices are:

$$\begin{bmatrix} 0 & 0 & 1/2 & \boxed{1/2} \\ 1/2 & \boxed{1/1} & 0 & 0 \\ 0 & 0 & \boxed{1/1} & 1/5 \\ \boxed{1/2} & 1/5 & 0 & 0 \end{bmatrix}_{t=1}$$

$$\begin{bmatrix} 0 & 0 & \boxed{1/1} & 0 \\ \boxed{1/1} & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & \boxed{1/4} \\ 0 & \boxed{1/4} & 0 & 1/2 \end{bmatrix}_{t=2}$$

$$\begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 1/1 & 0 & 0 \\ 1/3 & 0 & 1/1 & 0 \\ 1/3 & 0 & 0 & 1/1 \end{bmatrix}_{t=3}$$

Again, last matrix cannot be scheduled without missing a deadline. So the whole schedule is not feasible.

Lastly, consider matrix 3:

$$\begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1/2 & 1/1 & 0 & 0 \\ 1/2 & 0 & 1/1 & 0 \\ 0 & 1/5 & 0 & 1/1 \end{bmatrix}$$
$$t = 1$$

This matrix cannot be scheduled meeting all the deadlines. So, matrix 3 leads to a not feasible schedule.

Appendix C:

Algorithm EDD

$k \leftarrow 1$; $succ \leftarrow true$; $\tau \leftarrow 0$;

$\Delta \leftarrow$ l.c.m.(d_{ij}); $T_2 \leftarrow T$; $D_2 \leftarrow D$;

sort all the deadline occurrences, until Δ, and put them in *occur*;

while $\tau < \Delta$ **and** $succ$ **do**

begin

$list \leftarrow Sort[D_2]$;

$T_1 \leftarrow T$; $S_k \leftarrow 0$;

while $T_1 \neq 0$ **do**

begin

scan $list$ from the first entry,

until you find one with $t^1_{ij} > 0$;

set s^k_{ij} to t^1_{ij};

set row i and column j of T_1 to zero;

end;

find the minimum positive value sk_{min} in S_k;

set all non zero entries of S_k equal to sk_{min};

$T_2 \leftarrow T_2 - S_k$; $D_2 \leftarrow D_2 - sk_{min}$;

$\tau \leftarrow \tau + sk_{min}$;

if $\tau = \Delta - 1$

then Stop with $succ = true$

and $\{S_1, S_2, \ldots, S_m\}$ is the schedule for T and D

else if $d_{ij} - t_{ij} \geq 0$ for each $t_{ij} > 0$

then $k \leftarrow k + 1$

else $succ = false$;

if *occur*$\neq \emptyset$ **and** $\tau =$*occur*[1] **then**

begin

find d_{ij} such that τ **mod** $d_{ij} = 0$;

set $t^2_{ij} = t_{ij}$, $d^2_{ij} = d_{ij}$;

occur $\leftarrow$ *occur* - *occur*[1];

end

end;

References

[1] P. Barcaccia and M. A. Bonuccelli. A Polynomial Time Optimal Algorithm for Time Slot Assignment in Variable Bandwidth Systems. *ACM-IEEE Transactions on Networking*, 2(3):247–251, 1994.

[2] P. Barcaccia, M. A. Bonuccelli, and M. Di Ianni. Minimum Length Scheduling of Precedence Constrained Messages in Distributed Systems. In *EuroPar '96 Conference*, Lecture Notes in Computer Science. Springer Verlag, Berlin, 1996.

[3] G. Bongiovanni, D. Coppersmith, and C. K. Wong. An Optimal Time Slot Assignment for a SS/TDMA System with Variable Number of Trasponders. *IEEE Transaction on Communication*, 29:721–726, 1981.

[4] M. A. Bonuccelli and M. C. Clò. Scheduling of Real Time Messages in Optical Broadcast-and-Select Networks. manuscript, submitted for publication, available from: http://www.di.unipi.it/ bonucce/HRTsporadico.ps.gz.

[5] M. A. Bonuccelli and M. C. Clò. EDD Algorithm Performance Guarantee for Periodic Hard-Real-Time Scheduling in Distributed Systems. Technical report, Dipartimento di Informatica, Università di Pisa, 1999.

[6] M. A. Bonuccelli, I. S. Gopal, and C. K. Wong. Incremental Time Slots Assignment in SS/TDMA Satellite Systems. *IEEE Transaction on Communication*, 37:870–874, 1989.

[7] M. S. Borella and B. Mukherjee. Efficient Scheduling of Nonuniform Packet Traffic in a WDM/TDM Local Ligthwave Network with Arbitrary Transceiver Tuning Latencies. *IEEE J. on Selected Areas on Communication*, 14:923–934, 1996.

[8] H. Choi, H. A. Choi, and M. Azizoglu. Efficient Scheduling of Transmissions in Optical Broadcast Networks. *IEEE/ACM Transaction on Networking*, 4:913–920, 1996.

[9] J. Giles and B. Hajek. Scheduling Multirate Periodic Traffic in a Packet Switch. In *1997 Conference on Information Sciences and Systems at John Hopkins University*. available in postscript version at http://tesla.csl.uiuc.edu/ hajek/.

[10] I. S. Gopal, M. A. Bonuccelli, and C. K. Wong. Scheduling in Multibeam Satellites with Interfering Zones. *IEEE Transactions on Communication*, 31:941–951, 1983.

[11] R. Jain, K. Somalwar, J. Werth, and J. C. Browne. Heuristics for the Scheduling i/o Operations. *IEEE Transactions on Parallel and Distributed Systems*, 8(3):310–320, Mar. 1997.

[12] F. Jia, B. Mukherjee, and J. Iness. Scheduling Variable-Length Messages in Single-Hop Multichannel Local Ligthwave Networks. *IEEE/ACM Transaction on Networking*, 3(3):477–489, Aug. 1995.

[13] M. H. Klein, J. P. Lehoczky, and R. Rajkumar. Rate-Monotonic Analysis for Real-Time Industrial Computing. *IEEE COMPUTER*, pages 24–33, January 1994.

[14] J. P. Lehoczky. Real-Time Resource Management Techniques. In ed. J. J. Marciniak, editor, *Encyclopedia of Software Eng.*, pages 1001–1020. John Wiley and Sons, New York, 1994.

[15] J. Leung and J. Whitehead. On the Complexity of Fixed-Priority Scheduling of Periodic Real-Time Tasks. *Performance Evaluation*, 2(4):237–250, 1982.

[16] C. L. Liu and W. Layland. Scheduling Algorithms for Multiprogramming in a Hard Real-Time Enviroment. *J. ACM*, 20(1):46–61, Jan. 1973.

[17] I. R. Philp and J. W. Liu. A Switch Scheduling Problem for Real Time Periodic Messages. Submitted to Telecommunications Systems Journal, available in postscript version at http://pertsserver.cs.uiuc.edu/papers.

[18] G. R. Pieris and G. H. Sasaki. Scheduling Transmissions in WDM Broadcast-and-Select Networks. *IEEE/ACM Transaction on Networking*, 2:105–110, Apr. 1994.

[19] G. N. Rouskas and M. H. Ammar. Analysis and Optimization of Transmission Schedules for Single-Hop WDM Networks. *IEEE/ACM Transaction on Networking*, 3:211–221, Apr. 1995.

[20] G. N. Rouskas and V. Sivaraman. On the Design of Optimal TDM Schedules for Broadcast WDM Networks with Arbitrary Transceiver Tuning Latencies. In *Proc. IEEE INFOCOM'96*, pages 1217–1224, 1996.

[21] O. Serlin. Scheduling of Time Critical Processes. In *Proceedings of the Spring Joint Computer Conference*, pages 925–932, 1972.

[22] J. A. Stankovic, M. Spuri, M. Di Natale, and G. C. Buttazzo. Implications of Classical Scheduling Results for Real-Time Systems. *Computer*, pages 16–25, 1995.

[23] H. Y. Tyan, C. J. Hou, B. Wang, and C.C. Han. On Supporting Time-Constrained Communications in WDMA-based Star-coupled Optical Networks. In *Proc. IEEE Real Time Systems Symposium*, pages 175–184, 1996.