Comments on Window-Constrained Scheduling

Richard West *Member, IEEE* and Yuting Zhang

Abstract

This short report clarifies the behavior of DWCS with respect to Theorem 3 in our previously published paper [1], and describes an alternative approach to make guarantees for arbitrary window-constraints.

Index Terms – Real-time systems, multimedia, window-constraints, scheduling.

1 Introduction

In our previously published paper [1], Theorem 3 implies that the DWCS algorithm holds for *all* possible window-constraints. DWCS can satisfy Theorem 3 under certain conditions and the proof shows a particular case, when there are two classes of window-constraints x_i/y_i and x_j/y_j, such that $x_i = y_i - 1$, $x_j = y_j - 1$, and $x_j/y_j < x_i/y_i$. While DWCS can be shown to satisfy Theorem 3 when each stream (or, equivalently, job) J_i has a window-numerator $x_i = y_i - 1$, regardless of the value of y_i, the algorithm can fail to produce a feasible schedule for certain other window-constraints.

Without loss of generality we define a window-constraint on J_i as requiring "m_i out of k_i deadlines to be *met*", as opposed to allowing "x_i out of y_i deadlines to be *missed*" in non-overlapping windows of k_i deadlines (such that $k_i = y_i$ and $m_i = y_i - x_i$). Algorithms such as DWCS update the *current* window-constraint m_i'/k_i' of each job J_i by reducing m_i' by one each time a job is serviced in its current period, T_i, and also by reducing k_i' by one each time a new period starts. Further details regarding window-constraint adjustments can be found in our earlier work on DWCS [1]. We can now state the following theorem to show how it affects feasibility of DWCS for general window-constraints.

Theorem 1. *With respect to Theorem 3 in our original paper and given that we have* arbitrary *initial window-constraints: In each* non-overlapping *window of size q in the hyper-period, H, there* can *be more than q jobs out of n with* current *window-constraint $m_i'/k_i' = 1$ at any time, when $U_{min} = \sum_{i=1}^{n} \frac{m_i C_i}{k_i T_i} \leq 1$ ($C_i = 1,\ T_i = q, \forall i$).*

Proof. Suppose there are n jobs queued for service at time $t = 0$. Suppose also at time aq ($a < min_i(k_i), q < n$), there are $q+1$ jobs each with $m'_i = k'_i \neq 0$. Let Γ_j be the set of jobs that have been serviced for j instances in period $[0, aq)$, such that $|\Gamma_j| = n_j$. Out of each set Γ_j let $\bar{n_j}$ be the number of jobs whose current window-constraints are $m'_i = k'_i > 0$. Finally, let m_{ij}/k_{ij} be the initial window-constraint of each job J_i in Γ_j.

In the interval $[0, aq)$, each of the n_j jobs in $\Gamma_j \mid 0 \le j \le a$ changes its window-constraint as follows:

$$n_0: \ (m_{i0},\ k_{i0}) \stackrel{(0,aq)}{\longrightarrow} (m_{i0},\ k_{i0}-a) \ \vdash \ \bar{n_0}: \ m_{i0} = k_{i0} - a > 0$$
$$n_1: \ (m_{i1},\ k_{i1}) \stackrel{(1,aq)}{\longrightarrow} (m_{i1}-1,\ k_{i1}-a) \ \vdash \ \bar{n_1}: \ m_{i1} - 1 = k_{i1} - a > 0$$

$$n_a: \ (m_{ia},\ k_{ia}) \stackrel{(a,aq)}{\longrightarrow} (m_{ia}-a,\ k_{ia}-a) \ \vdash \ \bar{n_a}: \ m_{ia} - a = k_{ia} - a > 0$$

where, $\sum_{j=0}^{a} n_j = n; \ \ \sum_{j=0}^{a} \bar{n_j} = q+1, \ \ \sum_{j=0}^{a} jn_j \le aq.$

Now, consider the following case: $n_0 = \bar{n_0} = q+1, \ \bar{n_1} = \bar{n_2} = \cdots = \bar{n_a} = 0$. Then, for any job J_i, there must exist a value $b_j \mid b_j < a$, such that the jth instance of J_i is serviced in the interval $[b_j q, \ (b_j+1)q)$. Therefore, we have

$$\frac{m_{ij} - (j-1)}{k_{ij} - b_j} \ge \frac{m_{i0}}{k_{i0} - b_j} = \frac{k_{i0} - a}{k_{i0} - b_j}$$

$$\Rightarrow \ \frac{m_{ij}}{k_{ij}} \ge \frac{(k_{i0} - a)(k_{ij} - b) + (k_{i0} - b_j)(j-1)}{(k_{i0} - b_j)k_{ij}}$$

$$U_{min} = \frac{k_{i0} - a}{k_{i0} q}(q+1) + \sum_{j=1}^{a} \sum_{i=0}^{n_j} \frac{m_{ij}}{k_{ij} q} \le 1, \quad \sum_{j=0}^{a} jn_j \le aq$$

Given that a solution to the above inequalities exists (e.g., $a = 2, n_1 = 0, b_2 = 1, q = 4, k_{i0} = 3, m_{i0} = 1, n_0 = 5, n_2 = 4, m_{i2} = 35, k_{i2} = 64$), it must be that the theorem holds.

□

2 Pfair-based Virtual Deadline Scheduling (PVDS)

To satisfy Theorem 3 in our original paper [1] for *arbitrary* window-constraints, and hence show a feasible schedule is possible for 100% resource utilization (when $C_i = 1, T_i = q, \forall i$), we propose an approach called PVDS. First, we define the virtual deadline, $Vd_i(t)$, of a job J_i at any time t such that:

$$Vd_i(t) = t_{s_i} + (l_i + 1)\frac{k_i T_i}{m_i} \ \ (l_i = 0, 1, ..., m_i - 1)$$

where t_{s_i} is the start time of the current window of size k_iT_i, which is $\lfloor\frac{t}{k_iT_i}\rfloor k_iT_i$. l_i is the number of job instances that have already met their deadlines in the current window before t. In ordering jobs for service, the one with the earliest virtual deadline at time t has highest priority. If an instance of J_i is not serviced in its request period, T_i, its virtual deadline stays the same, whereas if a job instance is serviced, its virtual deadline increases by $\frac{k_iT_i}{m_i}$. If m_i instances of J_i are serviced in the current window, J_i is ineligible for service until the start of its next window, unless all other jobs have received their minimum service requirements.

Lemma 1. *There is no idle time before the failure of a synchronized job set where* $U_{min} = \sum_{i=1}^{n} \frac{m_iC_i}{k_iT_i} = 1$ $(C_i = 1, T_i = q, \forall i)$*, under pfair-based virtual deadline scheduling.*

Proof. Observe that a synchronized job set is one in which all the jobs in the set are queued for service at the same time. For simplicity, we can assume these jobs are all ready for service at time $t = 0$. Now, suppose $[t, t + a)$ is the first idle period before at least one of the jobs in the set fails to meet its service constraints. According to the algorithm, if there is idle time, each job J_i has either satisfied its window-constraint in the current window, or has finished service in the current period, $T_i = q$. Henceforth, let us define two sets of jobs: (1) Γ_0 is the set of jobs that have satisfied their window-constraints in the current window, by the start of the current period, and, (2) Γ_1 is the set of jobs that each have been serviced $l_i \mid (l_i{\leq}m_i)$ times in the current window, with the l_ith instance of J_i serviced in the current period.

Let J_{i_k} be the job J_i in the set Γ_k. For each and every job J_{i_0} in Γ_0, its virtual deadline (or start time of its next non-overlapping window) at time, t, satisfies the constraint $\lceil\frac{t}{k_{i_0}T_{i_0}}\rceil k_{i_0}T_{i_0}{\geq}t + a$. Similarly, since each and every job in Γ_0 has been serviced by the start of the current period, the previous virtual deadline of each job J_{i_0} must be less than the previous virtual deadline of each and every job, J_{i_1}, in Γ_1. In what follows, let $|\Gamma_0| = n_0$, $|\Gamma_1| = n_1$, and l_{i_1} be the number of instances of each job J_{i_1} in Γ_1. Therefore,

$$\forall\ i_0, i_1 \quad \lfloor\frac{t}{k_{i_1}T_{i_1}}\rfloor k_{i_1}T_{i_1} + \frac{k_{i_1}T_{i_1}}{m_{i_1}}l_{i_1} \geq \lceil\frac{t}{k_{i_0}T_{i_0}}\rceil k_{i_0}T_{i_0}$$

$$Let\ \ \lceil\frac{t}{k_pT_p}\rceil k_pT_p = max_{i_0}(\lceil\frac{t}{k_{i_0}T_{i_0}}\rceil k_{i_0}T_{i_0})$$

$$\Rightarrow \forall i_1, \quad \lfloor\frac{t}{k_{i_1}T_{i_1}}\rfloor k_{i_1}T_{i_1} + \frac{k_{i_1}T_{i_1}}{m_{i_1}}l_{i_1} \geq \lceil\frac{t}{k_pT_p}\rceil k_pT_p$$

$$\Rightarrow \lfloor\frac{t}{k_{i_1}T_{i_1}}\rfloor m_{i_1} + l_{i_1} \geq \lceil\frac{t}{k_pT_p}\rceil k_pT_p\frac{m_{i_1}}{k_{i_1}T_{i_1}} \tag{1}$$

$$\forall i_0, \ \lceil \frac{t}{k_{i_0}T_{i_0}} \rceil k_{i_0}T_{i_0} \geq t + a > t$$

$$t = \sum_{i=1}^{n_0} (\lceil \frac{t}{k_{i_0}T_{i_0}} \rceil m_{i_0}) + \sum_{j=n_0+1}^{n} (\lfloor \frac{t}{k_{i_1}T_{i_1}} \rfloor m_{i_1} + l_{i_1}) \quad (2)$$

$$From\ (1),\ (2) \Rightarrow t > \sum_{i=1}^{n_0} (\frac{t}{k_{i_0}T_{i_0}} m_{i_0}) + \sum_{j=n_0+1}^{n} (\lceil \frac{t}{k_pT_p} \rceil k_pT_p \frac{m_{i_1}}{k_{i_1}T_{i_1}})$$

$$> \sum_{i=1}^{n_0} (\frac{t}{k_{i_0}T_{i_0}} m_{i_0}) + \sum_{j=n_0+1}^{n} t\frac{m_{i_1}}{k_{i_1}T_{i_1}} = t(\sum_{i=1}^{n} \frac{m_i}{k_iT_i}) = t$$

This implies $t > t$ which is impossible, thereby yielding a contradiction to the supposition there is idle time before a failure. □

Lemma 2. *A job set is schedulable by pfair-based virtual deadline scheduling when* $U_{min} = \sum_{i=1}^{n} \frac{m_iC_i}{k_iT_i} = 1$ $(C_i = 1, T_i = q, \forall i)$.

Proof. Assume a schedule fails at time t for job J_i, where $t = ak_iT_i + bT_i$ $(0 \leq b < k_i, a \geq 0)$. Let l_i be the number of instances of job J_i serviced in the current window before t, so that $m_i - l_i - 1 = k_i - b$. Therefore,

$$\lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + \frac{k_iT_i}{m_i}(l_i + 1) - t = ak_iT_i + \frac{k_iT_i}{m_i}(m - k_i + b) - ak_iT_i - bT_i$$

$$= \frac{k_iT_i}{m_i}(m_i - k_i + b - \frac{m_i}{k_i}b) = \frac{k_iT_i}{m_i}(k_i(\frac{m_i}{k_i} - 1) + b(1 - \frac{m_i}{k_i}))$$

$$= \frac{k_iT_i}{m_i}(1 - \frac{m_i}{k_i})(b - k_i) \leq 0 \Rightarrow \lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + \frac{k_iT_i}{m_i}(l_i + 1) \leq t \quad (3)$$

Since job J_i fails at time t, and its virtual deadline is less than t, J_i is not serviced in the interval $[t - T_i, t)$. Now, let l_j be the number of instances of job J_j serviced in the current window before t. It follows that $T_i = q$ jobs other than J_i will be served in the interval $[t - T_i, t)$. Each of these other jobs, J_j, has its most recent virtual deadline at $\lfloor \frac{t}{k_jT_j} \rfloor k_jT_j + l_j \frac{k_jT_j}{m_j}$, which is less than or equal to job J_i's current virtual deadline at $\lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + (l_i + 1)\frac{k_iT_i}{m_i}$. Moreover, all jobs other than J_i must have most recent virtual deadlines that are less than or equal to J_i's current virtual deadline. If this were not the case, J_i would be able to service $l_i + 1$ instances before t. Therefore,

$$\forall j \neq i, \quad \lfloor \frac{t}{k_jT_j} \rfloor k_jT_j + l_j\frac{k_jT_j}{m_j} \leq \lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + (l_i+1)\frac{k_iT_i}{m_i}$$

$$\Rightarrow \lfloor \frac{t}{k_jT_j} \rfloor m_j + l_j \leq (\lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + (l_i+1)\frac{k_iT_i}{m_i})\frac{m_j}{k_jT_j} \tag{4}$$

From Lemma1, we know there is no idle before the first failure at time t. Therefore,

$$t = \lfloor \frac{t}{k_iT_i} \rfloor m_i + l_i + \sum_{j\neq i}(\lfloor \frac{t}{k_jT_j} \rfloor m_j + l_j) \tag{5}$$

$$From\ (4),\ (5) \Rightarrow t \leq \lfloor \frac{t}{k_iT_i} \rfloor m_i + l_i + \sum_{j\neq i}((\lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + (l_i+1)\frac{k_iT_i}{m_i})\frac{m_j}{k_jT_j})$$

$$\Rightarrow t \leq \lfloor \frac{t}{k_iT_i} \rfloor m_i + l_i + (\lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + (l_i+1)\frac{k_iT_i}{m_i})\sum_{j\neq i}\frac{m_j}{k_jT_j}$$

$$\Rightarrow t \leq \lfloor \frac{t}{k_iT_i} \rfloor m_i + l_i + (\lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + (l_i+1)\frac{k_iT_i}{m_i})(1 - \frac{m_i}{k_iT_i})$$

$$\Rightarrow t \leq \lfloor \frac{t}{k_iT_i} \rfloor k_iT_i + \frac{k_iT_i}{m_i}(l_i+1) - 1 \tag{6}$$

Equations (3) and (6) imply that $t+1 \leq t$, which is impossible, so the assumption that a schedule fails at time t cannot hold.

□

Extending the previous lemma, the following Theorem can be shown to hold (although we omit the proof for brevity):

Theorem 2. *There exists a feasible pfair-based virtual deadline schedule for a synchronized job set* Γ, *where* $U_{min} = \sum_{i=1}^{n} \frac{m_iC_i}{k_iT_i} \leq 1$ $(C_i = 1, T_i = q, \forall i)$.

References

[1] R. West, Y. Zhang, K. Schwan, and C. Poellabauer. Dynamic window-constrained scheduling of real-time streams in media servers. *IEEE Transactions on Computers*, 53(6):744–759, June 2004.