Utilization-Bound Based Schedulability Analysis of Weighted Round Robin Schedulers

Jianjia Wu
Department of Computer Science, Texas A&M University
jianjiaw@cs.tamu.edu

Jyh-Charn Liu
Department of Computer Science, Texas A&M University
liu@cs.tamu.edu

Wei Zhao
Department of Computer Science Rensselaer Polytechnic Institute
zhaow3@rpi.edu

Abstract

Schedulability analysis is a cornerstone of modern real-time scheduling theory development. Utilization-bound based schedulability test is considered one of most efficient and effective schedulability tests, because of its extremely low runtime overhead. Deriving the utilization bound for a given real-time system has, nevertheless, been a challenging task as it usually requires comprehensive modeling and understanding of the system and payload tasks. This paper is focused on deriving utilization bounds for weighted round robin schedulers. We demonstrate how to establish a unified modeling framework and then use it to derive utilization bounds. We obtain the optimal parameter selection that maximizes the utilization bound, then compare the new bound with those of fixed priority schedulers, and timed token ring schedulers. The new bound is further extended to systems with sporadic job requests. We argue that our modeling framework is highly versatile, and hence, can be easily tailored for analysis of other types of real-time systems.

1. INTRODUCTION

In this paper, we address issues related to utilization bound analysis of Weighted Round Robin (WRR) schedulers for hard real-time systems. By using a unified framework to analytically model the payload tasks and the scheduler, we derive utilization bounds under which the deadlines of the real-time tasks are guaranteed when a proper weight assignment algorithm is used.

Schedulability analysis has been a cornerstone of modern real-time scheduling theory development. Schedulability analysis is important for the design and operation of real world applications. Additionally, schedulability analysis of a given system can reveal much information about the system properties and capabilities, hence allowing improvement of the systems. Yet, a major challenge is how to devise a schedulability test with low complexity, so that the test can be effectively deployed in real world applications. The *utilization based schedulability* test [38] is one of most preferred schedulability tests, in which a new task can be admitted only if the total system utilization is lower than a pre-determined bound. The utilization bound based schedulability test is efficient and effective because of its extremely low runtime overhead and its provision of a safe operational margin [60].

Deriving the utilization bound for a given real-time system has, nevertheless, presented a challenge as it usually requires comprehensive modeling and understanding of the scheduler and payload tasks. Though this paper focuses on deriving utilization bounds for WRR schedulers, we stress that our goal is not to promote a specific scheduling algorithm, since it is obvious that each scheduling algorithm has its own advantages, disadvantages, and most appropriate application domains. Rather, we would like to demonstrate how to construct a unified modeling framework which can then be used to derive utilization bounds for WRR scheduling algorithms. Consequently, we argue that our modeling framework is highly versatile and can be used for analysis of other types of real-time systems.

In any case, WRR algorithms have broad real-world applications, such as process scheduling [12], [41], [48], [52], [57], load balancing of web server farm, [16], [34], server virtualization, etc. WRR is also the baseline algorithms used in bandwidth allocation [29], [30], [53], [59], e.g. Cisco Catalyst 5000 Family Switches [66], Cisco Switching Load Balancing [23]. Moreover, WRR and its variants have been proposed for wireless radio networks, e.g. (E)GPRS radio interface [31], Bluetooth protocol [15], [28], [36], wireless multimedia network [24], [25]. Finally, many embedded hardware real-time systems provide WRR discipline as a main scheduling implementation [9], [49].

In this paper, we will first establish a unified framework, following a similar approach in [60], to characterize payload tasks and WRR schedulers. Based on this framework, we are able to derive the utilization bounds for WRR schedulers. Different from the previous studies, our bound is parameterized by important system configuration factors, including (1) the number of tasks, (2) the normalized deadline (which measures the tightness of deadline assignment,) (3) tasks set workload burst-ness, (4) the overhead ratio, and (5) the normalized round robin frequency (i.e. the number of rotations in a time interval of the shortest relative deadline of the tasks). Consequently, our new bound allows one to easily determine the effects of system parameters on the utilization bounds. We will show how to obtain the optimal parameter selection that maximizes the bounds. We will also compare the bound of WRR schedulers with those of fixed priority schedulers, and timed token ring schedulers. We will further extend our results to systems with sporadic job requests, e.g. interrupt.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 introduces our system model. In Section 4, a parameterized, closed-form schedulability bound is derived for WRR system. Detailed comparisons of the newly derived bound with the known results of other schedulers are provided in Section 5. The bound is further extended to system with sporadic jobs in Section 6. Conclusions are provided in Section 7.

2. RELATED WORK

Utilization bound analysis can be traced back to the seminal work [38], in which the authors derived the well-known 69% bound for rate monotonic scheduler/deadline monotonic scheduler (RMS/DMS) on single processor systems, where relative deadlines of periodic tasks are equal to their periods. Since then, extensive research has been conducted on utilization bound analysis, including the generalized bound for arbitrary deadline assignment schemes in [32], [33], and [46], the generalization of the bound result to multiframe tasks [42], [43], and harmonic chain tasks [19]. Utilization bounds for other types of schedulers and tasks were also extensively studied, i.e. the timed token ring protocols [3], [39], [63], [64], [65], the network environment fixed priority scheduler [62], and fixed priority scheduler with non-periodic tasks [1], [2]. Generalization of the utilization bounds from single processor to multiple processors were studied in [1], [4], [7], [10], [20], [26], [37], [44], [51]. Schedulability analysis for hierarchy and composite systems were studied in [22], [55]. Alternative forms of schedulability testing that is not utilization bound based have proven interesting, and much research work has been done in [5], [8], [11], [13], [27], [35], [47], [59], and [60]. Recently, utilization bounds and alternative schedulability tests for system with various constraints are studied extensively, e.g. temperature constraints [58], precedence constraints [40], variable deadlines constraints [54], preemptive constraints [10] and fairness constraints [20].

Unlike the extensive work on utilization bound and schedulability analysis performed for various scheduling systems, little has been done for WRR, despite the broad adoption of WRR in commercial systems. It has been proven in [50] that with proper weight assignment, WRR schedulers can provide deadline guarantees for real-time computing. However, most existing work on WRR had focused on improving fairness [29], [30], [53], [56], optimizing weight assignment [31], reducing burst-ness in distributing services to tasks [6], [59], and others. Limited work has been done on how to support quality of service guarantee with efficient and simple schedulability test algorithms, e.g. utilization bound based testing, which is desirable in many large-scale systems. A general understanding on the utilization bound of the WRR and its sensitivity to system parameters remains an open problem and is a main focus of this paper.

3. SYSTEM MODEL

In this section, we establish a unified framework that can analytically model and represent the payload tasks and scheduler service.

3.1 Workload and Service Functions

Let $\Gamma = \{T_1, T_2, ..., T_n\}$ denote a task set in a single processor system, where T_i is the i^{th} task and the index i would be omitted in the later discussions when the context is clear. Each of the tasks is composed of a sequence of jobs and the worst-case execution time of a job is called its *job size*. A job can start its execution after its *release time*, t_r, and must be finished by its *absolute deadline* $t_d = t_r + D$ where D is called *relative deadline*. For a job, the time elapsed from the release time t_r to the completion time t_f is called the *delay* (response time) of the job, and the worst-case (i.e., largest) delay of all jobs in a task is denoted by d^*. The jobs of a task have the same relative deadline but may not have the same size. Jobs within a task are executed in a first come, first served order.

The resource demand of task T is denoted as $f(t)$, the *workload function* for T,

$$f(t) = \text{total size of jobs released from } T \text{ in } [0,\ t]. \quad (3.1)$$

Similarly, the processor time actually received by task T is denoted as $g(t)$, the *service function* for T,

$g(t) =$ CPU time rendered to jobs of task T in $[0,\ t]$. (3.2)

Based on the definitions of d^*, $f(t)$ and $g(t)$, the worst case delay can be easily derived [13]:

$$d^* = \sup_{t\geq 0}\left(\inf\left(\tau \mid f(t) \leq g(t+\tau)\right)\right). \tag{3.3}$$

sup(Q) is the smallest upper bound of Q and *inf*(Q) denotes the greatest lower bound of Q. It can be verified that $\inf\left(\tau \mid f(t) \leq g(t+\tau)\right)$ is the delay of the jobs arriving at time t. Thus d^* is the worst case delay of all the jobs from task T.

A basic function of the schedulability test is to determine whether or not the following inequality holds:

$$d^* \leq D. \tag{3.4}$$

One schedulability test technique is to calculate d^* using (3.3) and compare the result against D. Unfortunately, this method is usually unsuitable for online operation because exact forms of $f(t)$ and $g(t)$ may not be available when the schedulability test is made. Even if $f(t)$ and $g(t)$ are available, they are often too cumbersome to handle. A practical solution is to use some alternative forms of $f(t)$ and $g(t)$ for schedulability test.

3.2 Payload Task Model

3.2.1 Workload Constraint Function

Much work has been performed to find simple alternative representations of $f(t)$ available at schedulability test time. For example, a typical alternative is the *workload constraint function* $F(I)$ introduced in [21]. $F(I)$ is said to be a workload constraint function for task T if for any $0 \leq I \leq t$

$$f(t) - f(t-I) \leq F(I). \tag{3.5}$$

For any F that satisfies (3.5) $F(I)$ is an upper bound of total size of jobs released in the time window $[t-I,\ t]$. We use I in (3.5) because F is defined on the domain of time intervals, while $f(t)$ is defined on the domain of time. Note that a large group of functions can satisfy (3.5) and for convenience we focus on non-decreasing functions that satisfy $F(0) = 0$.

3.2.2 S-shaped Workload Constraint Functions

In [60], it was found that most real-time tasks could be modeled by the *s-shaped* workload constraint functions. As its name suggests, an s-shaped workload constraint function consists of segmented pieces, and resembles a staircase. The values of an s-shaped workload constraint function increase only at border points of segments. The segment length S is fixed and the increments may not be identical for the first L segments where L is a parameter. Formally, an s-shaped workload constraint function can be expressed as follows:

$$F(I) = \begin{cases} \sum_{j=1}^{h} C^j & h \leq L \\ \sum_{j=1}^{L} C^j + (h-L)C & h > L \end{cases}, \tag{3.6}$$

where $h = \lceil I/S \rceil$, C^j is the increment at the beginning of the j^{th} segment, and C is the constant increment after the L^{th} segment. When $L = 1$, an s-shaped constraint function reduces to the periodic task model $F(I) = \lceil I/P \rceil C$. In this paper, we assume

$$C^1 \geq C^2 \geq \cdots \geq C^L \geq C. \tag{3.7}$$

The s-shaped workload constraint function has a similar (stepwise) form as the multiframe function [42], but they are fundamentally different. The s-shaped function is a "bound" of the workload function, while the multiframe function is an exact representation of the workload. Using the workload bound for schedulability analysis gives us much more freedom in characterizing the schedulability conditions, as was demonstrated in [60] and [61]. More details on s-shaped workload constraint functions can be found in [60] and [61].

3.2.3 Parameters of Task Set

To capture the properties of the tasks, we introduce two task parameters in this section. The first parameter is the *normalized deadline* k_i for T_i which measures the tightness of task deadline:

$$k_i = D_i / S_i, \tag{3.8}$$

where D_i is the relative deadline of T_i and S_i is defined in (3.6). We follow the conventions in [19], [42], [33], [38] that for $i = 1,\ 2,\ ...,\ n$

$$k_i = k, \tag{3.9}$$

k can be viewed as the deadline using S as the measurement unit and characterizes tightness of task deadlines. In general, the smaller the k, the more difficult it is to schedule the tasks.

The second parameter, μ_i, is defined to characterize the *burst-ness* of a task T_i:

$$\mu_i = \frac{F(S_i)/S_i}{F(\lceil k \rceil S_i)/(\lceil k \rceil S_i)}, \tag{3.10}$$

and the *burst-ness* for the whole task set is defined as:

$$\mu = \max_{i=1,\ 2,\ ...,\ n}(\mu_i). \tag{3.11}$$

By (3.7), one can notice that

$$\mu \geq 1. \tag{3.12}$$

μ is ratio between the workload rate in a time window S_i, and that in $\lceil k \rceil S_i$. Larger μ denotes a more bursty workload.

3.3 Scheduler Model

Recall that *g* defined in (3.2) represents the amount of services a task may receive via a scheduler and, therefore, reflects the behavior of the scheduler. However, we cannot practically use the *g* defined in (3.2) since it may be too cumbersome, or unavailable. In this section, we consider the alternatives of g.

3.3.1 Service Constraint Function

A common alternative to $g(t)$ is the *generalized service constraint* function introduced in [14], [17], and [18]. $G(I)$ is said to be a *generalized service constraint function* if for any $t \geq 0$, there exists $I \leq t$ that preserves the property

$$g(t) \geq f(t-I) + G(I)\ . \tag{3.13}$$

Typically, $G(I)$ is assumed to be non-decreasing and $G(0) \geq 0$. (3.13) means that for any *t* we can find *I*, such that (1) all the jobs released in [0, $t-I$] have been served, and (2) for jobs released in [$t-I$, *t*], at least *G*(*I*) amount have been served. For more details and justifications of this definition, refer to [60] and [61].

3.3.2. Weighted Round Robin Scheduler

For a WRR scheduler, task services are time-multiplexed in a cyclic, round robin fashion. At its turn, task T_i can operate for up to H_i time unit where H_i is the time allocated to T_i. Typically, H_i is calculated as:

$$H_i = O_i \cdot \left(TCT - \tau\right), \tag{3.14}$$

where O_i, $0 \leq O_i \leq 1$, is the weight of T_i, and the TCT is the *target cycle rotation time*, which is the desired time to complete one round of service for all tasks, and τ is the time overhead for round robin operations in each round, i.e., the context switching costs in processor systems or the propagation delays in distributed systems. Generally speaking, τ can be expressed as:

$$\tau = n \cdot \tau_0\ , \tag{3.15}$$

where *n* is the number of tasks in the system and τ_0 is a *overhead constant*, e.g. the cost of one context switching.

3.3.3. Service Constraint Functions of WRR Schedulers

For the WRR scheduler, we have the following result on its service constraint function.

Theorem 1: For a WRR scheduler, its service constraint function for task T_i is given by

$$G_i(I) = \left\lfloor I/(n\tau_0 + \textstyle\sum_{j=1}^{n} H_j) \right\rfloor H_i. \tag{3.16}$$

Proof. Let t be an arbitrary time instant. If at t task T_i has at least one unfinished job, let s be the last time before t when T_i has no unfinished job, or

$$g_i(s) = f_i(s). \tag{3-17}$$

In time interval [s , t], the scheduler served at least $\left\lfloor I/(\tau + \sum_{j=1}^{n} H_j) \right\rfloor$ rounds with a serving time length of H_i each round. In other words, task T_i received at least $\left\lfloor I/\left(\tau + \sum_{j=1}^{n} H_j\right) \right\rfloor H_j$ amount of service. Formally, we have

$$g_i(t) - g_i(s) \geq \left\lfloor I/(\tau + \textstyle\sum_{j=1}^{n} H_j) \right\rfloor H_i. \tag{3-18}$$

By substituting (3-17) into (3-18), we have

$$g_i(t) - f_i(s) \geq \left\lfloor I/(\tau + \textstyle\sum_{j=1}^{n} H_j) \right\rfloor H_i. \tag{3-19}$$

By comparing (3-19) with (3.13), we known (3.16) is a service constraint function for the WRR scheduler.
If at t all the jobs from task T_i have been served, then by letting $s=t$, we can easily proof that (3.16) holds also. Q.E.D

We can make the following observation for (3.16):

- The service constraint functions are of periodic shapes, i.e. the values of the function increase only at multiples of $n\tau_0 + \sum_{j=1}^{n} H_j$ at the amount of H_i for T_i.
- The service constraint function is a decreasing function of the τ_0, but an increasing function of H_i.
- When τ_0 is zero and $TCT \to 0$, WRR becomes the Generalized Process Sharing(GPS) system and the service constraint function reduces to $G_i(I) = O_i \cdot I$.

3.3.3 Parameters of WRR Schedulers

Selection of proper system parameters is critical to precisely capture the system dynamics and to derive a useful bound. For this purpose, we will introduce two parameters in this section.

To measure the portion of time consumed in round robin operations, we define the *overhead ratio* α as follows:

$$\alpha = \tau_0 / TCT\ , \tag{3.20}$$

where τ_0 is the overhead constant defined in (3.15).

To capture the effect of target cycle rotation time on schedulability bounds, we define the *normalized rotation frequency* as follows:

$$\gamma = \left\lfloor D_{\min} / TCT \right\rfloor, \tag{3.21}$$

where

$$D_{\min} = \min(D_i) \, , \quad (3.22)$$

and we assume that

$$D_{\min} \ge TCT \, . \quad (3.23)$$

γ is the number of rounds the system rotates within the time interval D_{min}. The larger the γ, the faster the system rotates.

3.4 Schedulability Bound

With workload and service constraint functions defined in (3.5) and (3.13) for modeling task workload and scheduler service, a general schedulability test result is derived in [60].

Theorem 2 [60]**:** A task is schedulable if for any $I \ge 0$

$$F(I) \le G(I+D) \, . \quad (3.24)$$

where D is the relative deadline of the task.

Though one can use (3.24) for each task to decide its schedulability, it may be time consuming, since (3.24) needs to be checked for all $I \ge 0$. An alternative test is based on the schedulability bound. To perform this test, we need to generalize the classical definition of *utilization* for general tasks. Utilization is the resource consumption rate in a measuring time window. For periodic tasks, task period is typically used as the length of the measuring window, but this approach is not applicable to non-periodic tasks because they have no well defined "period." An option proposed in [1] and [4] is to use the task relative deadline as the measuring window, although we find that it is still too restrictive for the design of a versatile utilization bound analysis system for WRR. To relax the constraints, we propose to define the length of the measuring window as a linear scale of the relative deadline θD, where $\theta > 0$ is called the *scaling parameter* and D is the relative deadline of the task. Utilization defined on the basis of our proposed measurement window is called the *scaled workload rate*:

$$W_i(\theta) = F_i(\theta D_i) / \theta D_i \quad (3.25)$$

and the task set workload rate as follows:

$$W(\theta, \Gamma) = \sum_{i=1}^{n} W_i(\theta). \quad (3.26)$$

When the context is clear, the term "scaled" may be omitted. When $\theta = 1$, (3.26) reduces to the definition in [1] and [4]. The parameter θ enables flexible representation of workload for different scheduling and workload scenarios.

We say that $W^*(\theta)$ is a schedulability bound if an arbitrary task set Γ is schedulable when the following condition holds:

$$W(\theta, \Gamma) < W^*(\theta) \, . \quad (3.27)$$

Theorem 3: $W^*(\theta)$ is a schedulability bound if the following condition holds:

$$W^*(\theta) \le \inf_{\Gamma \in \Omega^*} \left(W(\theta, \Gamma) \right) . \quad (3.28)$$

where

$$\Omega^* = \left\{ \Gamma \mid \exists T_i \in \Gamma \text{ such that } F_i(I) > G_i(I+D) \right\} . \quad (3.29)$$

Proof: Let Ω_{ns} denote the collection of non-schedulable task sets. By definition, a schedulability bound is a lower bound of the workload rates of the task sets in Ω_{ns}. By Theorem 2, we have $\Omega_{ns} \subset \Omega^*$. Hence, any lower bound of the workload rate of the task sets in Ω^* is a schedulability bound. Q.E.D.

4. SCHEDULABILITY BOUNDS OF WRR SCHEDULERS

In this section, we analyze the schedulability bounds of WRR schedulers using the general system model introduced in the previous section.

To support hard real-time computing, O_i must be properly allocated. In this paper, we will show that an effective weight assignment scheme is the *normalized weight assignment:*

$$O_i = W_i(1) / W(1, \Gamma) \, , \quad (4.1)$$

where $W_i(1)$ and $W(1, \Gamma)$ are defined in (3.25) and (3.26). By (4.1) the weight assigned to a task is proportional to its workload rate. This scheme naturally matches many resource allocation requirements in practice, i.e. assigning larger weights to users with greater resource consumption rates.

4.1 Schedulability Bound of WRR Schedulers

By substituting the service constraint functions of the WRR scheduler derived in (3.16) into Theorem 3 and solving the resulting optimization problem, we have the follow Theorem.

Theorem 4: A schedulability bound with scaling parameter $\theta = \lceil k \rceil / k$ for WRR scheduler with normalized weight assignment and s-shaped tasks is given by

$$W^*\left(\lceil k \rceil / k\right) = 1/(1/\gamma + 1) \cdot \left(1 - n\alpha\right) \cdot 1/\mu \cdot \min\left(1, \, k\right) . (4.2)$$

Proof Sketch: For any non-schedulable task set, by (3.24), we know there exists a T_i and $I \ge 0$ such that $F_i(I) > G_i(I+D)$. By (3.16) we know that $F_i(I) > \lfloor (I+D_i)/TCT \rfloor \cdot F_i(D_i)/D_i \cdot (TCT - \tau)/W(1, \Gamma)$ and thus $W(1, \Gamma) > \lfloor (I+D_i)/TCT \rfloor \cdot F_i(D_i)/D_i \cdot (TCT - \tau)$. By (3.6), we have $W(1, \Gamma) = W\left(\lceil k \rceil / k, \Gamma\right) \lceil k \rceil / k$ thus $W(\lceil k \rceil / k, \Gamma)$ is greater than $k / \lceil k \rceil \cdot \lfloor (I+D_i)/TCT \rfloor \cdot F_i(D_i)/D_i \cdot (TCT - \tau)$ which is lower bound by $1/(1/\gamma + 1) \cdot \left(1 - n\alpha\right) \cdot 1/\mu \cdot \min\left(1, \, k\right)$. That is, for any non-schedulable task set, $W(\lceil k \rceil / k, \Gamma)$ is greater than $1/(1/\gamma+1)\cdot(1-n\alpha)\cdot 1/\mu \cdot \min(1, k)$ and thus the theorem. Complete steps are given in Appendix A.

The bound presented in (4.2) is a parameterized expression which allows obtaining bounds for different systems by plugging in specific values. In the next subsections, we will discuss the sensitivity of the bound and compare it with other schedulers.

4.2 Sensitivity Analysis

By (4.2), we can make the following observations on the sensitivity of the schedulability bound:

- For given *n*, *k*, μ, and α, the bound increases with the cycle rotation frequency. Thus, decreasing *TCT* improves the chance for a task set to be scheduled. Figure 1 illustrates this trend for $\alpha = 0$.
- For given *n*, *k*, μ, and α, the bound decreases with the task set burst-ness μ. Increasing μ implies a more bursty workload, which tends to be more difficult to schedule than less bursty ones. The bound is maximized when μ =1, which corresponds to periodic tasks.
- For given *n*, *k*, γ, and μ, the bound increases when overhead ratio α decreases. The bound is maximized when α =0, which is an ideal case in which the operation overhead is zero.
- For given k, γ, μ, and α, the bound decreases with the number of tasks and approaches zero when $n \to 1/\alpha$, i.e., all processor time is consumed by round robin operation.
- For periodic tasks with relative D_i=P_i, and $\gamma = 2$, (two rounds of rotation per D_{min} interval), the bound is 67%. When the normalized deadline increases from 1.0 to 10.0, the bound remains at 67%. This is illustrated as a serial of points in Figure 2. Note that this does not imply that *k* has no effect on the schedulability test, since the workload rate is measured differently.
- For periodic tasks with D_i=P_i, and $\gamma = 1$, (one round of rotation per D_{min} interval), the bound is 50%. This is highlighted on Figure 3 as one point.

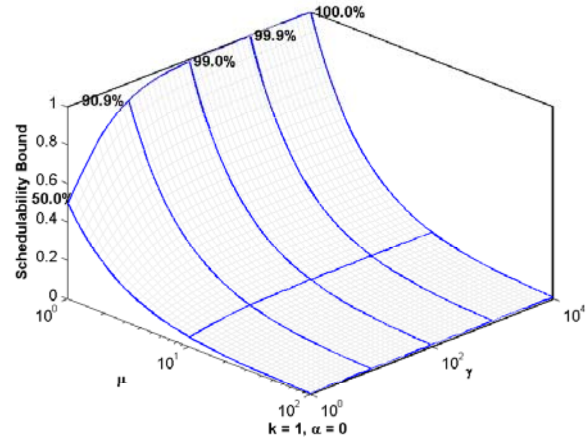

Figure 1. Schedulability Bound with $k = 1$

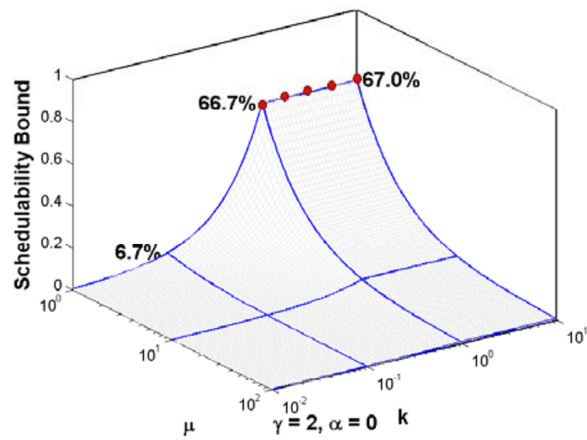

Figure 2. Schedulability Bound with $\gamma = 2$

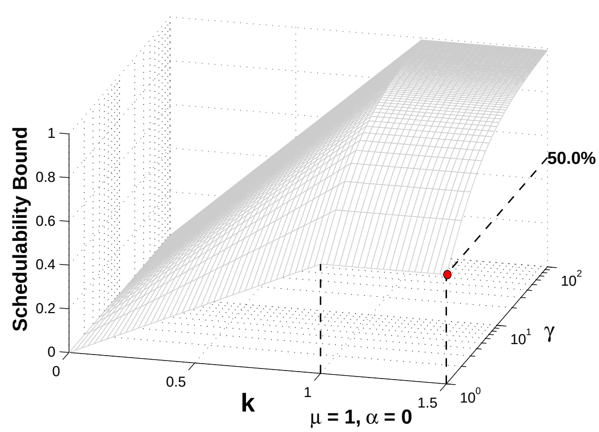

Figure 3. Schedulability Bound with $\mu = 1$

4.3 Optimal *TCT* Selection

In a WRR system, parameters *n*, *k* and μ are fixed for a given task set. One can try to increase its schedulability bound by adjusting *TCT*. By (3.20) and

(3.21) we know that setting larger *TCT* may reduce the operational overhead ratio but, at the same time lead to lower rotation frequency. Thus, *TCT* should be adjusted to balance the operational overhead ratio and the rotation frequency. The following theorem gives the optimal *TCT* value that maximizes the schedulability bound.

Theorem 5: The schedulability bound of a WRR scheduler for a given set of tasks is maximized when the value of *TCT* is selected as follows:

$$TCT = D_{\min} / \gamma^* , \quad (4\text{-}3)$$

where

$$\gamma^* = \begin{cases} 1 & \text{if } n\tau_0 / D_{\min} \ge 1/3 \\ \left\lfloor \sqrt{D_{\min}/n\tau_0 + 1} \right\rfloor - 1 & \text{if } n\tau_0 / D_{\min} < 1/3 \text{ and } Z\left(\left\lfloor \sqrt{D_{\min}/n\tau_0+1}\right\rfloor - 1\right) < Z\left(\left\lceil \sqrt{D_{\min}/n\tau_0+1}\right\rceil - 1\right) \\ \left\lceil \sqrt{D_{\min}/n\tau_0 + 1} \right\rceil - 1 & \text{otherwise} \end{cases} , \quad (4\text{-}4)$$

and

$$Z(x) = 1/(1/x+1) \cdot (1 - n\tau_0 x / D_{\min}) . \quad (4\text{-}5)$$

When the *TCT* takes the optimal value, the schedulability bound is given by

$$W^*(\lceil k \rceil / k) = \frac{1}{1/\gamma^* + 1} \cdot \left(1 - \frac{n\tau_0}{D_{\min}} \gamma^*\right) \cdot \frac{1}{\mu} \cdot \min(1, k) . \quad (4\text{-}6)$$

Proof: Please see Appendix B. Q.E.D.

By a careful observation of Theorem 5, we notice that the optimal *TCT* value only depends on $n\tau_0 / D_{\min}$, which is essentially the percentage of time consumed by round robin operations within a window of length $D_{\min}$. When $n\tau_0 / D_{\min} \ge 1/3$, the optimal *TCT* equals to $D_{\min}$ and when $n\tau_0 / D_{\min} < 1/3$, the optimal *TCT* value would be either $D_{\min} / (\lfloor \sqrt{D_{\min}/n\tau_0 + 1} \rfloor - 1)$ or $D_{\min} / (\lceil \sqrt{D_{\min}/n\tau_0 + 1} \rceil - 1)$.

Figure 4 plots the trend of the schedulability bound with the optimal *TCT* selection for the case of $k = 1$ and $\mu = 1$. It is clear that, when the optimal value of *TCT* is used, the schedulability bound is a monotonic decreasing function of $n\tau_0 / D_{\min}$. The rate of decrease is low for small values of $n\tau_0 / D_{\min}$, say, less than 0.01, and gradually increases when $n\tau_0 / D_{\min}$ becomes larger. When $n\tau_0 / D_{\min}$ approaches 1.0, the schedulability bound is close to 0%.

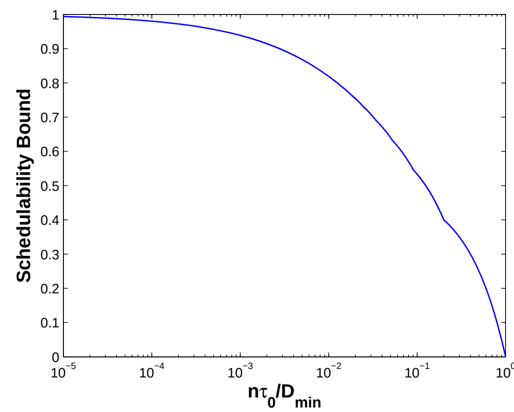

Figure 4. Schedulability Bound with the Optimal *TCT*

5. COMPARISON WITH EXISTING RESULTS

We now compare the newly derived schedulability bound of WRR with those of fixed priority schedulers and the timed token ring schedulers.

5.1 Comparison with Fixed Priority Schedulers

For simplicity, we focus on periodic tasks and assume that overhead ratio α of the weighted round robin schedule is 0.

Consider a set of periodic tasks with deadlines equal to their periods. A schedulability bound for the rate monotonic scheduler is 69% [38]. For the same task system and WRR scheduler, we know that $\mu = 1$ and $k = 1$. By substituting these parameters into (4.2), we have a schedulability bound of $\gamma / (\gamma + 1)$. The curve of this function in plotted in Figure 5. As illustrated by Figure 5, the bound of the weighted round robin scheduler is lower than the bound of fixed priority schedulers when $\gamma < 2.26$ and out-performs the rate monotonic scheduler when $\gamma > 2.26$.

For a set of periodic tasks with deadlines being half as long as their periods, a schedulability bound for the rate monotonic scheduler is 50% [33]. For the same workload, under the WRR scheduler we know that $\mu = 1$, and k=1/2. By substituting these parameters into (4.2), we have a bound of $\gamma / (2(\gamma + 1))$. Since $\gamma / (2(\gamma + 1)) \le 1/2$, we know that the rate monotonic scheduler outperforms the WRR scheduler in this case.

For a set of periodic tasks with deadlines being twice as long as their periods, a schedulability bound for the rate monotonic scheduler is 81% [33]. For the same task system with the WRR scheduler, we know that $\mu = 1$, and k=2. By substituting these parameters into (4.2), we have a bound of $\gamma / (\gamma + 1)$. It is evident that when $\gamma = 4.32$, the WRR scheduler achieves the same bound as the rate monotonic scheduler and out-performs it when γ further increases.

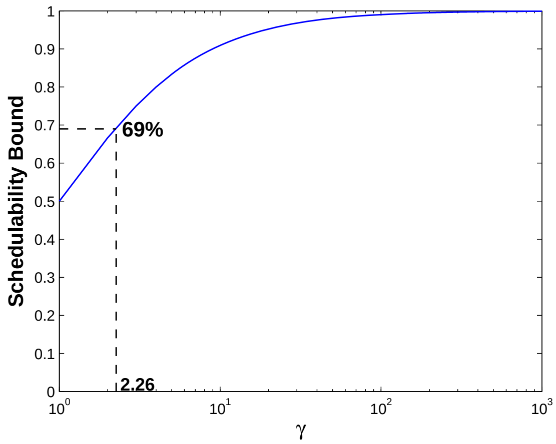

Figure 5. Schedulability Bound with $\mu = 1$

Based on the above analysis, we know that for the periodic tasks, the weighted round robin scheduler can achieve the same or even higher schedulability bound than the rate monotonic schedulers. This is because that fixed priority scheduler renders service to a job only if there is no higher priority job in the queue, even though completing the lower priority job first may avoid missing a deadline. In other words, fixed priority scheduler may allocate more service time to high priority tasks than what is needed to guarantee the deadlines, while WRR scheduler assigns task service time in proportional to its workload rate, which avoids over-allocation of service time to certain tasks. As such, WRR scheduler can outperform the rate monotonic scheduler under certain conditions. We should note that the above analysis focuses on a very simple case in which tasks are of a periodic shape and round robin overhead is negligible. However, similar trends still hold for more complex cases.

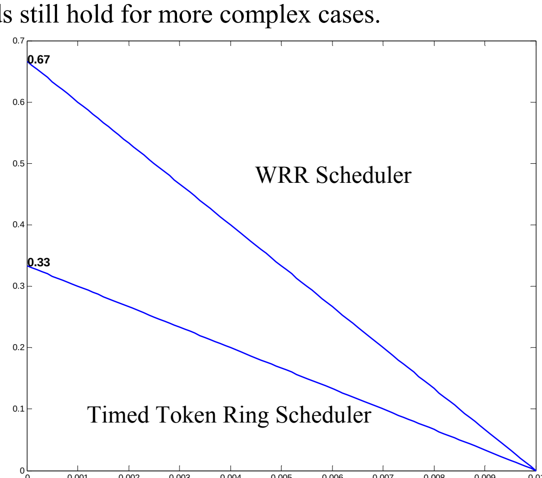

Figure 5. Schedulability Bound Comparison between WRR and Timed Token Ring Scheduler

5.2 Comparison with Timed Token Protocol

A local area network protocol closely related to WRR scheduler is the timed token ring scheduler of FDDI networks [3], [63], [65]. In FDDI, communication nodes are interconnected as a ring topology. Each node has real-time and non real-time packets. Real-time packets have hard deadlines, e.g. packets must be sent before their deadlines, while non-real-time packets do not have any deadline. Similar to the weighted round robin scheduler, a token is rotated among nodes and the desired time to finish one round of a token rotation is called the target token rotation time (*TTRT).* Upon receiving the token, a node will first send its real-time packets up to the allocated amount of time. Each node also keeps track of the last token rotation time, denoted by *TTR.* If the token arrives earlier in the last round, i.e. *TTR* < *TTRT*, then it will send its non-real-time packets up to the *TTRT-TRT* amount of time. The rationale behind this is to "steal" the unused time slots in the last token rotation.

Due to the interference of the non real-time packets, the service available to each node may be less than that provided by a weighted round robin scheduler. In turn, the schedulability bound will be lower. It has been proven in [3] that for periodic tasks with relative deadlines equal to their periods and normalized weight assignment scheme with a token rotation at least twice that of any interval of length of minimum task periods, the timed token ring scheduler has a schedulability bound of $(1-n\alpha)/3$. For this system, by (3.21), (3.8), and (3.10), we know that k=1, μ =1, and γ =2. By substituting them into (4.2), we know that a schedulability bound of the weighted round robin scheduler is $2(1-n\alpha)/3$, twice that of the timed token ring. Figure 6 illustrates this difference for the case of n=100.

6. EXTENSION FOR INTERRUPT SUPPORT

Timely service of sporadic jobs (e.g. interruptions) is an important issue for any real-time scheduler. The WRR scheduler can perform these jobs in two ways: *polling* based and *interrupts* based. In the first approach, a polling service task T_p can be added to the existing task set and scheduled in same way as other task sets. New sporadic job will be queued in a job queue managed by T_p and may be served when T_p receives the processor resource. The schedulability analysis for WRR with a polling service task is relatively simple and the bound derived in Theorem 5 can be directly used.

In systems that request immediate service of interrupts, an interrupt service task T_s can be added to the existing task set in a WRR system. T_s has the highest priority among all the tasks and will preempt any running task if there is a new sporadic job. We use C_s to denote the maximum amount of processor time required by the sporadic jobs released within any time interval of length P_s. In other words, the workload of T_s can be modeled by a workload constraint function $F_s = \lfloor I / P_s \rfloor C_s$. To model the context switching overheads, we assume that up to q sporadic jobs can be released within any time interval of length P_s. The context switching overheads for each sporadic job is, at most, $2\tau_0$ (switching from T_i to T_s and then from T_s to T_i). For convenience, we define

$$C_s' = C_s + 2q\tau_0 , \qquad (6\text{-}1)$$

C_s' can be considered the total worst case execution time spent on the sporadic jobs and related context switching for jobs released within any time interval of length P_s. We further define

$$\gamma' = \left\lfloor \left(D_{\min} - 2C_s' \right) / TTRT \right\rfloor \qquad (6\text{-}2)$$

γ' is the number of rotations within the period of $D_{\min} - 2C_s'$. By comparing (6-2) with (3.21), we can see that γ' takes into account the effect of the high priority interrupt jobs whose period is of minimum relative deadline. Now, we discuss the service constraint function of the tasks.

Theorem 6: A service constraint function provided by WRR system to task T_i, i=1, 2, …, n, is

$$G_i(I) = \left\lfloor \left(I - \lceil I / P_s \rceil C_s' \right) / \left(\tau + \sum_{j=1}^{n} H_j \right) \right\rfloor H_i \qquad (6\text{-}3)$$

Proof: Please see Appendix C. Q.E.D.

Theorem 7: A schedulability bound for WRR with scaling parameter $\theta = \lceil k \rceil / k$, normalized weight assignment, s-shaped tasks, and an interrupt service is given by

$$W^*(\lceil k \rceil / k) \geq \left(1 - n\alpha\right) \cdot \frac{1}{1/\gamma' + 1} \frac{1}{\mu} \min\left((1 - \frac{C_s'}{P_s}),\ k \cdot (1 - \frac{2C_s'}{D_{\min}}) \right)$$

(6-4)

Proof: Please see Appendix D. Q.E.D.

By a close observation of (6-4), we see that, for WRR system with interrupt service, both the workload rate of the interruptions and their worst case execution times (plus context switching overhead) affect the bound. The higher the workload of the interrupt service task (the more time spent on interrupt handling), the lower the bound. The ratio between C_s' and $D_{\min}$ also affects the bound. When there are no interrupts in the system, the bound reduces to that of the WRR system, derived in Theorem 5.

7. FINAL REMARKS

Deriving the utilization bound for a given real-time system has been a challenging task that requires comprehensive modeling and understanding of the scheduler and payload tasks. Much work on utilization bounds has been done for priority driven systems. Our work reported here is among the first to systematically derive the schedulability bound for WRR systems. The utilization bound we derived is a general one and parameterized for number of tasks n, round robin overhead ratio α , normalized rotation ratio γ , the normalized deadline k , and the task set workload burstness μ . From this general formula, one can easily obtain schedulability bounds for specific WRR system configuration by simple plugging in of proper parameters. We compared the bound for WRR system with fixed priority systems and timed-token ring systems.

A salient feature of our work is that the analysis is based on a unified modeling framework. As such, we argue that the modeling framework is highly versatile and hence can be tailored for analysis of other types of real-time systems. For example, a similar approach can be taken to analyze the schedulability bound of the well known Weighted Fair Queuing (WFQ) [23] scheduler since it is an approximation of the GPS scheduler and provides a similar service guarantee as WRR. Another interesting direction to consider is the schedulability bound of hierarchy scheduling policies, e.g. schedulers that integrate the fixed priority scheduling with the WRR.

REFERENCES

[1] T. Abdelzaher, B. Andersson, J. Jonsson, V. Sharma, "The Aperiodic Multiprocessor Utilization Bound for Liquid Tasks," in Proc. the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium, Washington DC, September 2002, pp. 173-184.

[2] T. Abdelzaher and C. Lu, "Schedulability analysis and utilization bounds for highly scalable real-time services," *Proc. 7th Real-Time Technology and Applications Symposium*, Taipei, Taiwan, 2001, pp. 15-25.

[3] G. Agrawal, B. Chen, W. Zhao, and S. Davari, "Guaranteeing synchronous message deadlines with the timed token protocol," *IEEE Trans. Computers*, vol. 43, no. 3, pp. 327–339, Mar. 1994.

[4] B. Andersson, "Static-priority scheduling on multiprocessors," Ph.D. dissertation, Dept. Computer Engineering., Chalmers Univ. of Technology, Göteborg, Sweden, 2003.

[5] B. Andersson, , Cecilia Ekelin, "Exact Admission-Control for Integrated Aperiodic and Periodic Tasks," *Proc. Real Time and Embedded Technology and Applications Symposium,* San Francisco, California, March 2005, pp. 76-85

[6] A. Francini, F. M. Chiussi, R. T. Clancy, K. D. Drucker, N. E. Idirene, “Enhanced Weighted Round Robin Schedulers for Bandwidth Guarantees in Packet Networks,” *proc. Quality of Service in Multiservice IP Networks: International Workshop, QoS-IP 2001*, Rome, Italy, January 2001, pp. 2005

[7] T. P. Baker, “Multiprocessor EDF and deadline monotonic schedulability analysis,” *Proc. 24th IEEE Int. Real-Time Systems Symposium*, Cancun, Mexico, 2003, pp. 120-129.

[8] T. P. Baker, M. Cirinei, “A Necessary and Sometimes Sufficient Condition for the Feasibility of Sets of Sporadic Hard-Deadline Tasks,” *Proc. the 27th IEEE International Real-Time Systems Symposium*, Washington, December 2006, pp. 178-190.

[9] F. Balarin, L. Lavagno, P. Murthy, “Scheduling for embedded real-time systems,” *Design & Test of Computers, IEEE Publication*, Mar 1998, Vol. 15, no.1, pp 71-82.

[10] S. Baruah, “Feasibility Analysis of Preemptive Real-Time Systems upon Heterogeneous Multiprocessor Platforms,” *Proc. Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International, Lisbon, Portugal,* December 2004, pp. 37-46.

[11] E. Bini, “Schedulability analysis of periodic fixed priority systems,” *IEEE Trans. Computers,* vol. 53, no.11, pp. 1462-1473, Nov. 2004.

[12] J. Bruno, E. Gabber, B. Ozden, A.Silberschatz, “The Eclipse Operating System: Providing Quality of Service via Reservation Domains,” *Proc. Annual Tech. Conf., USENIX*, June 1998, pp. 235 - 246.

[13] E. Bini, G. C. Buttazzo, and G. Buttazzo, “Rate monotonic analysis: the hyperbolic bound,” *IEEE Trans. Computers*, vol 52, no. 7, pp. 933-942, Jul. 2003.

[14] J. Y. Le Boudec and P. Thiran, *Network Calculus, a Theory of Deterministic Queuing Systems for the Internet*, New York: Springer-Verlag, 2001.

[15] A. Capone, M. Gerla, R. Kapoor, “Efficient polling schemes for Bluetooth picocells,” *Communications, 2001. ICC 2001. IEEE International Conference on*, Vol. 7, pp. 1990-1994, 2001.

[16] E. Casalicchio, S. Tucci, “Static and Dynamic Scheduling Algorithms for Scalable Web Server Farm,” *Proc. Ninth Euromicro Workshop on Parallel and Distributed Processing (PDP '01)*, Mantova, Italy, Feb. 2001, pp. 369-376.

[17] S. Chang, “On deterministic traffic regulation and service guarantee: a systematic approach by filtering,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 1096-1107, Aug. 1998.

[18] S. Chang, *Performance Guarantees in Communication Networks*, New York: Springer-Verlag, 2000.

[19] D. Chen, A. K. Mok, and T.-W. Kuo, “Utilization bound revisited,” *IEEE Trans. Computers*, vol. 52, no. 3, pp. 351-361, Mar. 2003.

[20] H. Cho, B. Ravindran, E. D. Jensen, “An Optimal Real-Time Scheduling Algorithm for Multiprocessors,” *Proc. Real-Time Systems Symposium, 2006. RTSS '06. 27th IEEE International, Dec. 2006,* Rio de Janeiro, Brasil, pp. 101-110.

[21] R. L. Cruz, "A calculus for network delay, part I: network elements in isolation," IEEE Transactions on Information Theory, 37(1), pp. 114–131, Jan. 1991.

[22] R. I. Davis, A. Burns, “Hierarchical Fixed Priority Pre-Emptive Scheduling,” *Proc. the 26th IEEE International Real-Time Systems Symposium,* Tucson, Arizona, December 2005, pp. 389-398.

[23] R. A. Deal , *Cisco Router Firewall Security,* Cisco Press, 2004

[24] H. Fattah, C. Leung, “An overview of scheduling algorithms in wireless multimedia networks,” *Proc. Wireless Communications, IEEE*, Oct. 2002, vol. 9, no. 5, pp. 76- 83.

[25] C. Fragouli, V. Sivaraman, S. Srivastava, “Controlled multimedia wireless link sharing via enhancedclass-based queuing with channel-state-dependent packet scheduling,” *Proc. INFOCOM '98, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies IEEE*, San Francisco, CA, Apr 1998, vol. 2, pp. 572-580.

[26] S. Funk, J. Goossens, S. Baruah, “On-Line Scheduling on Uniform Multiprocessors,” *Proc. Real-Time Systems Symposium, 2001. (RTSS 2001). Proceedings. 22nd IEEE*, London, UK, Dec. 2001, pp. 183-192.

[27] W. Hawkins, Tarek Abdelzaher, “Towards Feasible Region Calculus: An End-to-End Schedulability Analysis of Real-Time Multistage Execution,” *Proc. Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International*, Miami, Florida, December 2005, pp. 75-86.

[28] M. Kalia, D. Bansal, R. Shorey, “MAC scheduling and SAR policies for Bluetooth: a master driven TDDpico-cellular wireless system,” *Proc. Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop on*, San Diego, California, November, 1999, pp. 384-388.

[29] S.S. Kanhere, H. Sethu, A.B. Parekh, “Fair and efficient packet scheduling using Elastic Round Robin,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 3, pp. 324-336, Mar 2002

[30] M. Katevenis, S. Sidiropoulos, C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1265-1279, Oct. 1991

[31] A. Kuurne, A.P. Miettinen, “Weighted round robin scheduling strategies in (E)GPRS radio interface,” *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, Sept. 2004, vol. 5, pp. 3155- 3159.

[32] J. P. Lehoczky, “Fixed priority scheduling of periodic task sets with arbitrary deadlines,” *in Proc. 11th IEEE Real Time Systems Symposium*, Piscataway, NJ, 1990, pp. 201-209.

[33] J. P. Lehoczky and L. Sha, “Performance of real-time bus scheduling algorithms,” ACM SIGMETRICS Performance Evaluation Review, vol. 14, no. 1, May 1986, pp. 44–53.

[34] R. Levy, J. Nagarajarao, G. Pacifici, A. Spreitzer, A. Tantawi, A. Youssef, “Performance management for cluster based Web services”, *Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on*, March 2003, page(s): 247- 261

[35] X. Liu, T. Abdelzaher, “On Non-Utilization Bounds for Arbitrary Fixed Priority Policies,” *Proc. the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06),* San Jose, California, Apr. 2006, pp. 167-178.

[36] Y. Liu Q. Zhang W. Zhu, “A priority-based MAC scheduling algorithm for enhancing QoS support in Bluetooth

piconet," *Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on*, Chengdu, China, July 2002, vol. 1, pp. 544- 548.

[37] L. Lundberg, "Global Multiprocessor Scheduling of Aperiodic Tasks using Time-Independent Priorities," *Proc. Real-Time and Embedded Technology and Applications Symposium, 2003,* Washington, DC, May 2003, pp. 170-180.

[38] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM,* vol. 20, no. 1, pp. 46–61, Jan. 1973.

[39] N. Malcolm and W.Zhao, "Guaranteeing synchronous messages with arbitrary deadline constraints in an FDDI network," *Proc. the IEEE Conference on Local Computer Networks,* Minneapolis, MN, 1993, pp. 186-195.

[40] L. Mangeruca, A. Ferrari, "Uniprocessor scheduling under precedence constraints," *Proc the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06),* San Jose, California, Apr. 2006, pp. 157-166.

[41] Microsoft, Load-Balanced Cluster, http://msdn2.microsoft.com/en-us/library/ms978730.aspx, retrieved on May 22 2007

[42] A. K. Mok and D. Chen, "A multiframe model for real-time tasks," IEEE Trans. Software Engineering, pp. 635-645, vol.23, no.10, Oct. 1997.

[43] K. Mok and D. Chen, "A general model for real-time tasks," *Technical Report: CS-TR-96-24*, University of Texas at Austin.

[44] D. Oh and T. P. Bakker, "Utilization bounds for n-processor rate monotone scheduling with static processor assignment," *Real Time Systems J.*, vol. 15, no. 1, pp. 183-193, Nov. 1998.

[45] A. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[46] D.-T. Peng and K.G. Shin, "A new performance measure for scheduling independent real-time tasks," Journal of Parallel Distributing Computing, vol. 19, no. 12, pp. 11–16, 1993.

[47] R. Pellizzoni, S. Superiore S. Anna, "Improved Schedulability Analysis of Real-Time Transactions with Earliest Deadline Scheduling," *Proc. the 11th IEEE Real Time on Embedded Technology and Applications Symposium*, San Francisco, California , Mar. 2005, pp. 66-75

[48] qdisc, http://tldp.org/HOWTO/Traffic-Control-HOWTO/components.html, retrieved on May 22, 2007.

[49] A. Radulescu, J. Dielissen, K. Goossens, E. Rijpkema, P. Wielage, "an efficient on-chip network interface offering guaranteed services, shared-memory abstraction, and flexible network configuration," Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings, Vol. 2, Vol.2, pp. 878- 883, Feb. 2004

[50] A. Raha, N. Malcolm, Wei Zhao, "Hard real-time communications with weighted round robin service in ATM local area networks," *iceccs*, First IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'95), pp. 96-103, Aug. 1995.

[51] S Ramamurthy, "Scheduling Periodic Hard Real-Time Tasks with Arbitrary Deadlines on Multiprocessors," *Proc. the 23rd IEEE Real-Time Systems Symposium*, Austin, TX, Dec. 2002, pp. 59-68.

[52] M.J. Rutten, J.T.J. van Eijndhoven, E.D. Pol Egbert, G.T. Jaspers, P. van der Wolf, O.P. Gangwal, A. Timmer, "Eclipse: heterogeneous multiprocessor architecture for flexible media processing," *Proc. Parallel and Distributed Processing Symposium., International, IPDPS 2002*, Aug 2002, pp. 39-50.

[53] D. Saha, S. Mukherjee, S.K. Tripathi, "Carry-over round robin: a simple cell scheduling mechanism for ATM networks," IEE/ACM Transactions on Networking, Vol. 6, no. 6, pp. 779-796, Dec 1998

[54] C.-S. Shih, L. Sha, J. Liu, "Scheduling Tasks With Variable Deadlines," *Proc. Real-Time Technology and Applications Symposium, 2001*, London, UK, December 2002, pp. 120-122.

[55] I. Shin, I. Lee, "Compositional Real-Time Scheduling Framework," *Proc. the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, Lisbon, Portugal, December 2004, pp. 57-67.

[56] M. Shreedhar, G. Varghese, "Efficient fair queueing using deficit round robin," SIGCOMM Comput. Commun., Vol. 25, no. 4, pp. 231-242, Oct. 1995

[57] Technical Note TN2028, http://developer.apple.com/technotes/tn/tn2028.html, retrieved on May 22, 2007

[58] S. Wang, Riccardo Bettati, "Delay Analysis in Temperature-Constrained Hard Real-Time Systems with General Task Arrivals," *Proc. the 27th IEEE International Real-Time Systems Symposium,* Rio de Janeiro, Brazil, December 2006, pp. 323-334.

[59] Y.-T. Wang, T.-P. Lin, K.-C. Gan, "An improved scheduling algorithm for weighted round-robin cell multiplexing in an ATM switch," Comput. & Commun., Communications, 1994. ICC 94, SUPERCOMM/ICC '94, Conference Record, Serving Humanity through Communications. IEEE International Conference on, vol.2, pp. 1032-1037, May 1994.

[60] J. Wu, J.-C. Liu and W. Zhao, "On schedulability bounds of static priority schedulers," *Proc.* 11th IEEE Real-Time and Embedded Technology and Applications Symposium, San Francisco, CA, Mar, 2005, pp. 529-540.

[61] J. Wu, "General schedulability bound Analysis and its applications in real-time systems", Ph.D. dissertation, Dept. Computer Science, Texas A&M University, College Station, Texas, 2006.

[62] D. Xuan, C. Li, R. Bettati, J. Chen, and W. Zhao, "Utilization-based admission control for real-time applications," *Proc. 2000 Int. Conf. Parallel Processing*, Toronto, Canada, 2000, pp. 251–262.

[63] S. Zhang and A. Burns, "An optimal synchronous bandwidth allocation scheme for guaranteeing synchronous message deadlines with the timed-token MAC protocol," *IEEE/ACM Trans. Networking*, vol. 3, no. 6, pp. 729-741, 1995.

[64] S. Zhang , A. Burns, A. Mehaoua, E. S. Lee, and H. Yang, "Testing the schedulability of synchronous traffic for the timed token medium access control protocol,", *Real-Time Systems*, vol.22, no.3, pp.251-280, May 2002.

[65] Q. Zheng and K. G. Shin, "Synchronous bandwidth allocation in FDDI networks," *IEEE Transactions on Parallel and Distributed Systems*, vol.6 no. 12, pp.1332-1338, Dec. 1995.

[66] S Vegesna, *IP Quality of Service*, Cisco Press, 2001

APPENDIX A

Theorem 4: A schedulability bound with scaling parameter $\theta = \lceil k \rceil / k$ for weighted round robin scheduler with normalized weight assignment, and s-shaped tasks is given by

$$W^*(\lceil k \rceil / k) = 1/(1/\gamma + 1) \cdot (1 - n\alpha) \cdot 1/\mu \cdot \min(1,\ k) . \quad \text{(A-1)}$$

Proof. By Theorem 3, we known that a service constraint function provided to task T_i by a WRR scheduler is:

$$G_i(I) = \left\lfloor I / \left(\tau + \sum_{j=1}^{n} H_j \right) \right\rfloor H_i . \quad \text{(A-2)}$$

By (3.14) and (4.1) we have

$$H_i = (TTRT - \tau) \cdot W_i(1) / W(1,\ \Gamma) \quad \text{(A-3)}$$

By substituting (A-3) into (A-2), we have

$$G_i(I) = \lfloor I / TCT \rfloor \cdot W_i(1) / W(1,\ \Gamma) \cdot (TCT - \tau). \quad \text{(A-4)}$$

By Theorem 2, we know that a schedulability bound for WRR with scaling parameter $\theta = \lceil k \rceil / k$ is

$$W^*(\lceil k \rceil / k) = \inf_{\Gamma \in \Omega^*} \left(W(\lceil k \rceil / k,\ \Gamma) \right) . \quad \text{(A-5)}$$

where

$$\Omega^* = \{ \Gamma \mid \exists T_i \in \Gamma \text{ such that } F_i(I) > G_i(I + D) \} . \quad \text{(A-6)}$$

By substituting (A-4) into (A-6) and rearranging the resulting equation, we know that a schedulability bound is

$$W^*(\lceil k \rceil / k) = \inf_{\Gamma \in \Omega^*} \left(W(\lceil k \rceil / k,\ \Gamma) \right) . \quad \text{(A-7)}$$

where

$$\Omega^* = \{ \Gamma \mid \exists T_i \in \Gamma \text{ such that } W(1,\ \Gamma) > \lfloor (I + D_i) / TCT \rfloor \cdot W_i(1) / F_i(I) \cdot (TCT - \tau) \} . \quad \text{(A-8)}$$

$\forall \Gamma \in \Omega^*$, let us define

$$Z(i) = \min_{I \ge 0} \left(\lfloor (I + D_i) / TCT \rfloor \cdot (TCT - \tau) / F_i(I) \cdot W_i(1) \right) . \quad \text{(A-9)}$$

Then by (A-8), we have

$$W(1,\ \Gamma) > \min_{i = 1,\ 2,\ \ldots,\ n} \left(Z(i) \right) . \quad \text{(A-10)}$$

By substituting (3.25) and (3.20) into (A-9), we have

$$Z(i) = (1 - n\alpha) \cdot \min_{I \ge 0} \left(\frac{\lfloor (I + D_i) / TCT \rfloor}{(I + D_i) / TCT} \frac{I + D_i}{F_i(I)} \frac{F_i(D_i)}{D_i} \right) . \quad \text{(A-11)}$$

Since $(I + D_i) / TCT \le \lfloor (I + D_i) / TCT \rfloor + 1$, we have

$$Z(i) \ge (1 - n\alpha) \cdot \min_{I \ge 0} \left(\frac{\lfloor (I + D_i) / TCT \rfloor}{\lfloor (I + D_i) / TCT \rfloor + 1} \frac{I + D_i}{F_i(I)} \frac{F_i(D_i)}{D_i} \right) . \quad \text{(A-12)}$$

It is easy to verify that

$$\frac{\lfloor (I + D_i) / TCT \rfloor}{\lfloor (I + D_i) / TCT \rfloor + 1} = \frac{1}{1 + \frac{1}{\lfloor (I + D_i) / TCT \rfloor}} \ge \frac{1}{1 + \frac{1}{\lfloor D_{\min} / TCT \rfloor}} = \frac{1}{1 + 1/\gamma} . \quad \text{(A-13)}$$

where $D_{\min}$ and γ are defined in (3.22) and (3.21), respectively. By substituting (A-13) into (A-12), we have

$$Z(i) \ge (1 - n\alpha) \cdot \min_{I \ge 0} \left(1/(1 + 1/\gamma) \cdot (I + D_i) / F_i(I) \cdot F_i(D_i) / D_i \right) . \quad \text{(A-14)}$$

Let $I = mS_i + \omega$ where $0 \le \omega < S_i$, S_i is the segment length of the s-shaped workload constraint function defined in (3.6), and m is a non-negative integer. By (3.6), we have

$$F_i(I) \le F_i((m + 1) S_i) . \quad \text{(A-15)}$$

Substituting (A-15) into (A-14) and rearranging it, we get

$$Z(i) \ge (1 - n\alpha) \cdot \frac{1}{1/\gamma + 1} \min_{I \ge 0} \left(\frac{mS_i + \omega + D_i}{F_i((m + 1) S_i)} \frac{F_i(D_i)}{D_i} \right) . \quad \text{(A-16)}$$

Since $\omega \ge 0$, we have

$$Z(i) \ge (1 - n\alpha) \cdot \frac{1}{1/\gamma + 1} \min_{I \ge 0} \left(\frac{mS_i + D_i}{F_i((m + 1) S_i)} \frac{F_i(D_i)}{D_i} \right) . \quad \text{(A-17)}$$

By (3.6) and (3.7), it can be verified that

$$F_i((m + 1) S_i) / (m + 1) S_i \le F_i(S_i) / S_i . \quad \text{(A-18)}$$

Substituting (A-18) into (A-17) and rearranging it, we get

$$Z(i) \ge (1 - n\alpha) \cdot \frac{1}{1/\gamma + 1} \min_{I \ge 0} \left(\frac{mS_i + D_i}{(m + 1) F_i(S_i)} \frac{F_i(D_i)}{D_i} \right) . \quad \text{(A-19)}$$

By definition of s-shaped workload constraint function in (3.6) and k in (3.9), we know that

$$F_i(D_i) = F_i(kS_i) = F_i(\lceil k \rceil S_i) . \quad \text{(A-20)}$$

By substituting (A-20) into (A-19), we have

$$Z(i) \ge (1 - n\alpha) \cdot \frac{1}{1/\gamma + 1} \min_{I \ge 0} \left(\frac{mS_i + D_i}{(m + 1) F_i(S_i)} \frac{F_i(\lceil k \rceil S_i)}{kS_i} \right) . \quad \text{(A-21)}$$

By substituting (3.8) into (A-21) and rearranging the resulting inequality, we have

$$Z(i) \ge (1 - n\alpha) \cdot \frac{1}{1/\gamma + 1} \min_{I \ge 0} \left(\frac{m + k}{(m + 1) F_i(S_i)} \frac{F_i(\lceil k \rceil S_i)}{k} \right) . \quad \text{(A-22)}$$

Rewriting (A-22) into

$$Z(i) \ge (1 - n\alpha) \frac{1}{1/\gamma + 1} \frac{1}{k} \frac{F_i(\lceil k \rceil S_i)}{F_i(S_i)} \min_{m = 0,\ 1,\ 2,\ \ldots} \left(\frac{m + k}{m + 1} \right) . \quad \text{(A-23)}$$

It can be verified that

$$\min_{m = 0,\ 1,\ 2,\ \ldots} \left((m + k) / (m + 1) \right) \ge \min(1,\ k) . \quad \text{(A-24)}$$

By substituting (A-24) into (A-23), we have

$$Z(i) \ge (1 - n\alpha) \cdot \frac{1}{k} \cdot \frac{1}{1/\gamma + 1} \frac{F_i(\lceil k \rceil S_i)}{F_i(S_i)} \min(1,\ k) . \quad \text{(A-25)}$$

By (3.6) and (3.7), we have

$$F_i(\lceil k \rceil S_i) / F_i(S_i) = \lceil k \rceil / \mu_i \ge \lceil k \rceil / \mu . \quad \text{(A-26)}$$

By substituting (A-26) into (A-25), we get

$$Z(i) \ge (1 - n\alpha) \cdot \frac{1}{1/\gamma + 1} \cdot \frac{\lceil k \rceil}{k\mu} \cdot \min(1,\ k) . \quad \text{(A-27)}$$

By substituting (A-27) into (A-10), we have

$$W(1,\ \Gamma) \ge (1 - n\alpha) \cdot \frac{1}{1/\gamma + 1} \frac{\lceil k \rceil}{k\mu} \min(1,\ k) . \quad \text{(A-28)}$$

By (3.6), we know that

$$W(1,\ \Gamma)=\sum_{i=1}^{n}\frac{F_i(D_i)}{D_i}=\frac{\lceil k\rceil}{k}\sum_{i=1}^{n}\frac{F_i(\lceil k\rceil S_i)}{\lceil k\rceil S_i}=\frac{\lceil k\rceil}{k}W\left(\frac{\lceil k\rceil}{k},\ \Gamma\right). \quad \text{(A-29)}$$

By substituting (A-29) into (A-28) and rearranging the resulting inequality, we have

$$W(\lceil k\rceil/k,\ \Gamma)\geq(1-n\alpha)\cdot\frac{1}{1/\gamma+1}\cdot\frac{1}{\mu}\cdot\min(1,\ k). \quad \text{(A-30)}$$

Since (A-30) is true $\forall\Gamma\in\Omega^*$, we know that

$$\inf_{\Gamma\in\Omega^*}(W(\lceil k\rceil/k,\ \Gamma))\geq(1-n\alpha)\frac{1}{1/\gamma+1}\frac{1}{\mu}\min(1,\ k). \quad \text{(A-31)}$$

Then, by substituting (A-31) into (A-5), we get

$$W^*(\lceil k\rceil/k)\geq(1-n\alpha)\cdot\frac{1}{1/\gamma+1}\cdot\frac{1}{\mu}\cdot\min(1,\ k). \quad \text{(A-32)}$$

Comparing (A-32) with (A-1), we have proven the theorem. Q.E.D.

APPENDIX B

Theorem 5: The schedulability bound of a WRR scheduler, for a given set of tasks, is maximized when the value of *TCT* is selected as follows:

$$TCT=D_{\min}/\gamma^*, \quad \text{(B-1)}$$

where

$$\gamma^*=\begin{cases}1 & \text{if } n\tau_0/D_{\min}\geq 1/3\\ \left\lfloor\sqrt{D_{\min}/n\tau_0+1}\right\rfloor-1 & \text{if } n\tau_0/D_{\min}<1/3 \text{ and } Z\left(\left\lfloor\sqrt{D_{\min}/n\tau_0+1}\right\rfloor-1\right)<Z\left(\left\lceil\sqrt{D_{\min}/n\tau_0+1}\right\rceil-1\right)\\ \left\lceil\sqrt{D_{\min}/n\tau_0+1}\right\rceil-1 & \text{otherwise}\end{cases},$$

(B-2)

and

$$Z(x)=1/(1/x+1)\cdot(1-n\tau_0 x/D_{\min}). \quad \text{(B-3)}$$

When the *TCT* takes the optimal value, the schedulability bound is given by

$$W^*(\lceil k\rceil/k)=\frac{1}{1/\gamma^*+1}\cdot\left(1-\frac{n\tau_0}{D_{\min}}\gamma^*\right)\cdot\frac{1}{\mu}\cdot\min(1,\ k). \quad \text{(B-4)}$$

Proof: By (4.2), we know that a schedulability bound is

$$W^*(\lceil k\rceil/k)=\frac{1}{1/\gamma+1}\cdot(1-n\alpha)\cdot\frac{1}{\mu}\cdot\min(1,\ k). \quad \text{(B-5)}$$

Let us define

$$\omega=D_{\min}-\gamma\cdot TCT. \quad \text{(B-6)}$$

where γ is defined in (3.21). By (3.21), we have

$$\omega\geq 0. \quad \text{(B-7)}$$

By substituting (B-6) and (3.21) into (B-5), we have

$$W^*(\lceil k\rceil/k)=\frac{1}{1/\gamma+1}\cdot\left(1-\frac{n\tau_0}{(D_{\min}-\omega)/\gamma}\right)\cdot\frac{1}{\mu}\cdot\min(1,\ k). \quad \text{(B-8)}$$

It is evident that (B-8) will be maximized when $\omega=0$ and the maximum will be

$$W^*(\lceil k\rceil/k)=\frac{1}{1/\gamma+1}\cdot\left(1-\frac{n\tau_0}{D_{\min}}\gamma\right)\cdot\frac{1}{\mu}\cdot\min(1,\ k). \quad \text{(B-9)}$$

Now, we need to find a value of γ, $\gamma=1,\ 2,\ \ldots$, that maximizes (B-9). Clearly, the value of γ that maximizes (B-9) will also maximize

$$Z(\gamma)=\frac{1}{1/x+1}\cdot\left(1-\frac{n\tau_0}{D_{\min}}\gamma\right). \quad \text{(B-10)}$$

We can rewrite (B-10) as follows:

$$Z(\gamma)=\left(1-\frac{1}{\gamma+1}\right)\cdot\left((1+\frac{n\tau_0}{D_{\min}})-\frac{n\tau_0}{D_{\min}}(\gamma+1)\right). \quad \text{(B-11)}$$

An equivalent form of (B-11) is given by:

$$Z(\gamma)=1+2\frac{n\tau_0}{D_{\min}}-\frac{n\tau_0}{D_{\min}}\left((\frac{D_{\min}}{n\tau_0}+1)\frac{1}{\gamma+1}+(\gamma+1)\right). \quad \text{(B-12)}$$

Since $D_{\min}\geq TTRT\geq n\tau_0$, by calculating the derivation of (B-12) over γ, we know that (B-12) will be maximized when $\gamma=\sqrt{D_{\min}/n\tau_0+1}-1$. But since γ can only take the value of a positive integer, we know that (B-12) will attain its maximum either at

$$\gamma_0=\max\left(1,\ \left\lfloor\sqrt{D_{\min}/n\tau_0+1}\right\rfloor-1\right), \quad \text{(B-13)}$$

or

$$\gamma_1=\left\lceil\sqrt{D_{\min}/n\tau_0+1}\right\rceil-1. \quad \text{(B-14)}$$

That is, the optimal value of *TCT* that maximizes the schedulability bound is:

$$TCT=\begin{cases}D_{\min}/\gamma_0 & \text{if } Z(\gamma_0)\leq Z(\gamma_1)\\ D_{\min}/\gamma_1 & \text{otherwise}\end{cases}. \quad \text{(B-15)}$$

It is easy to verify that when $D_{\min}/n\tau_0\leq 3$, $\gamma_0=\gamma_1=1$ and thus (B-15) is equivalent to

$$TCT=D_{\min}/\gamma^*, \quad \text{(B-16)}$$

where

$$\gamma^*=\begin{cases}1 & \text{if } D_{\min}/n\tau_0\leq 3\\ D_{\min}/\gamma_0 & \text{if } D_{\min}/n\tau_0>3 \text{ and } Z(\gamma_0)<Z(\gamma_1)\\ D_{\min}/\gamma_1 & \text{otherwise}\end{cases}. \quad \text{(B-17)}$$

By substituting (B-16) into (B-9), we know that the maximum schedulability bound

$$W^*(\lceil k\rceil/k)=\frac{1}{1/\gamma^*+1}\cdot\left(1-\frac{n\tau_0}{D_{\min}}\gamma^*\right)\cdot\frac{1}{\mu}\cdot\min(1,\ k). \quad \text{(B-18)}$$

is attained when $TCT=D_{\min}/\gamma^*$. Hence, the theorem follows. Q.E.D

APPENDIX C

Theorem 6: A service constraint function provided by WRR system to task T_i, i=1, 2, …, n, is

$$G_i(I)=\left\lfloor\left(I-\lceil I/P_s\rceil C_s'\right)/\left(\tau+\sum_{j=1}^{n}H_j\right)\right\rfloor H_i \quad \text{(C-1)}$$

Proof: Given t and let s be the latest time instant before s when the system has no unfinished jobs. In $[s,\ t]$ there are at most $F_s(t-s)=\lfloor (t-s)/P_s \rfloor C_s$ amount of sporadic jobs. Since the total processing time in $[s,\ t]$ is $t-s$ and the sporadic jobs can consume up to $\lceil (t-s)/P_s \rceil C_s'$ amount of service, we known that at least $t-s-\lceil (t-s)/P_s \rceil C_s'$ amount of processor time has been rendered to regular tasks $T_1,\ \ldots,\ T_n$. There are at least $\left\lfloor \left(I-\lceil I/P_s \rceil C_s'\right)/\left(\tau+\sum_{j=1}^{n} H_j\right) \right\rfloor$ rounds in a WRR system. Since T_i receives at least H_i units of service time, in $[s,\ t]$, T_i receives at least $\left\lfloor \left(I-\lceil I/P_s \rceil C_s'\right)/\left(\tau+\sum_{j=1}^{n} H_j\right) \right\rfloor H_i$ amount of service. By definition of service constraint function, the theorem holds. Now, we will derive the schedulability bound for WRR system with interrupt services. Q.E.D.

APPENDIX D

Theorem 7: A schedulability bound with scaling parameter $\theta=\lceil k \rceil/k$, normalized weight assignment, s-shaped tasks, and an interrupt service is given by

$$W^*(\lceil k \rceil/k) \ge (1-n\alpha)\cdot\frac{1}{1/\gamma'+1}\frac{1}{\mu}\min\left((1-\frac{C_s'}{P_s}),\ k\cdot(1-\frac{2C_s'}{D_{\min}})\right) \quad \text{(D-1)}$$

Proof: By Theorem 6, we known that a service constraint function provided to task T_i is:

$$G_i(I)=\left\lfloor (I-\lceil I/P_s \rceil C_s')/(\tau+\sum_{j=1}^{n} H_j) \right\rfloor H_i \quad \text{(D-2)}$$

By (4.1) and (3.14), we have

$$H_i=(TCT-\tau)\cdot W_i(1)/W(1,\ \Gamma) \quad \text{(D-3)}$$

By substituting (D-2) into (D-3), we have

$$G_i(I)=\left\lfloor \frac{I-\lceil I/P_s \rceil C_s'}{\tau+\sum_{j=1}^{n} H_j} \right\rfloor H_i \frac{W_i(1)}{W(1,\ \Gamma)}(TCT-\tau) \quad \text{(D-4)}$$

By Theorem 2, we know that a schedulability bound for WRR with scaling parameter $\theta=\lceil k \rceil/k$ is

$$W^*(\lceil k \rceil/k)=\inf_{\Gamma\in\Omega^*}\left(W(\lceil k \rceil/k,\ \Gamma)\right). \quad \text{(D-5)}$$

where

$$\Omega^*=\left\{\Gamma \mid \exists T_i\in\Gamma \text{ such that } F_i(I)>G_i(I+D)\right\}. \quad \text{(D-6)}$$

By substituting (D-4) into (D-6) and rearranging the resulting equation, we know that a schedulability bound is

$$W^*(\lceil k \rceil/k)=\inf_{\Gamma\in\Omega^*}\left(W(\lceil k \rceil/k,\ \Gamma)\right). \quad \text{(D-7)}$$

where

$$\Omega^*=\left\{\Gamma \mid \exists T_i\in\Gamma \text{ such that } W(1,\ \Gamma)>\left\lfloor \frac{I-\lceil I/P_s \rceil C_s'}{\tau+\sum_{j=1}^{n} H_j} \right\rfloor \frac{W_i(1)}{F_i(I)}(TCT-\tau)\right\}. \quad \text{(D-8)}$$

For $\forall\Gamma\in\Omega^*$, let us define

$$Z(i)=\min_{I\ge 0}\left(\left\lfloor \frac{I+D_i-\lceil (I+D_i)/P_s \rceil C_s'}{\tau+\sum_{j=1}^{n} H_j} \right\rfloor H_i\frac{TCT-\tau}{F_i(I)}W_i(1)\right). \quad \text{(D-9)}$$

We can rewrite (D-9) into

$$Z(i)=\min_{I\ge 0}\left(\left\lfloor \frac{I+D_i-(\frac{I+D_i}{P_s}+1)C_s'}{TCT} \right\rfloor \frac{TCT-\tau}{F_i(I)}W_i(1)\right). \quad \text{(D-10)}$$

An equivalent form of (D-10) is

$$Z(i)=\min_{I\ge 0}\left(\left\lfloor \frac{(I+D_i)(1-C_s'/P_s)-C_s'}{TCT} \right\rfloor \frac{TCT-\tau}{F_i(I)}W_i(1)\right) \quad \text{(D-11)}$$

Then by (D-8), we have

$$W(1,\ \Gamma)>\min_{i=1,\ 2,\ \ldots,\ n}\left(Z(i)\right). \quad \text{(D-12)}$$

Le us define

$$x=\left((I+D_i)(1-C_s'/P_s)-C_s'\right)/TCT \quad \text{(D-13)}$$

By substituting (D-13), (3.25) , (3.20) into (D-12), we have

$$Z(i)=(1-n\alpha)\cdot\min_{I\ge 0}\left(\frac{\lfloor x \rfloor}{x}\frac{x}{F_i(I)}\frac{F_i(D_i)}{D_i}\right). \quad \text{(D-14)}$$

Since $x\le\lfloor x \rfloor+1$, we have

$$Z(i)\ge(1-n\alpha)\cdot\min_{I\ge 0}\left(\frac{\lfloor x \rfloor}{\lfloor x \rfloor+1}\frac{x}{F_i(I)}\frac{F_i(D_i)}{D_i}\right). \quad \text{(D-15)}$$

It is easy to verify that

$$\frac{\lfloor x \rfloor}{\lfloor x \rfloor+1}=\frac{1}{1+1/\lfloor x \rfloor}\ge\frac{1}{1+1/\left\lfloor \frac{D_{\min}-2C_s'}{TTRT} \right\rfloor}=\frac{1}{1+1/\gamma'}. \quad \text{(D-16)}$$

where $D_{\min}$ and γ' are defined in (3.22) and (6-2), respectively. By substituting (D-16) into (D-15), we have

$$Z(i)\ge(1-n\alpha)\cdot\min_{I\ge 0}\left(\frac{1}{1+1/\gamma'}\frac{(I+D_i)(1-C_s'/P_s)-C_s'}{F_i(I)}\frac{F_i(D_i)}{D_i}\right). \quad \text{(D-17)}$$

Let $I=mS_i+\omega$ where $0\le\omega<S_i$, S_i is the segment length of the s-shaped workload constraint function defined in (3.6), and m is a non-negative integer. By (3.6), we have

$$F_i(I)\le F_i((m+1)S_i). \quad \text{(D-18)}$$

Substituting (D-18) into (D-17) and rearranging it, we get

$$Z(i)\ge(1-n\alpha)\cdot\frac{1}{1/\gamma'+1}\min_{I\ge 0}\left(\frac{(mS_i+\omega+D_i)(1-C_s'/P_s)-C_s'}{F_i((m+1)S_i)}\frac{F_i(D_i)}{D_i}\right). \quad \text{(D-19)}$$

Since $\omega\ge 0$, we have

$$Z(i)\ge(1-n\alpha)\cdot\frac{1}{1/\gamma'+1}\min_{I\ge 0}\left(\frac{(mS_i+D_i)(1-C_s'/P_s)-C_s'}{F_i((m+1)S_i)}\frac{F_i(D_i)}{D_i}\right). \quad \text{(D-20)}$$

By (3.6) and (3.7), it can be verified that

$$F_i\left((m+1)S_i\right)/(m+1)S_i \le F_i\left(S_i\right)/S_i \,. \qquad \text{(D-21)}$$

Substituting (D-21) into (D-20) and rearranging it, we get

$$Z(i) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\min_{I\ge 0}\left(\frac{\left(mS_i+D_i\right)\left(1-C_s'/P_s\right)-C_s'}{(m+1)F_i\left(S_i\right)}\frac{F_i(D_i)}{D_i}\right) \,. \qquad \text{(D-22)}$$

By definition of s-shaped workload constraint function in (3.6) and k in (3.9), we know that

$$F_i(D_i) = F_i(kS_i) = F_i(\lceil k\rceil S_i) \,. \qquad \text{(D-23)}$$

By substituting (D-23) into (D-22), we have

$$Z(i) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\min_{I\ge 0}\left(\frac{\left(mS_i+D_i\right)\left(1-C_s'/P_s\right)-C_s'}{(m+1)F_i\left(S_i\right)}\frac{F_i(\lceil k\rceil S_i)}{kS_i}\right). \qquad \text{(D-24)}$$

Rewrite (A-22) into

$$Z(i) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\frac{F_i(\lceil k\rceil S_i)}{kF_i\left(S_i\right)}\min_{I\ge 0}\left(\frac{\left(m+k\right)\left(1-C_s'/P_s\right)-C_s'/S_i}{(m+1)}\right) \,. \qquad \text{(D-25)}$$

An equivalent form is

$$Z(i) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\frac{F_i(\lceil k\rceil S_i)}{kF_i\left(S_i\right)}\min_{I\ge 0}\left(\frac{\left(m+k\right)\left(1-C_s'/P_s\right)-kC_s'/D_i}{(m+1)}\right). \qquad \text{(D-26)}$$

Rearrange into

$$Z(i) \ge \frac{1-n\alpha}{1/\gamma'+1}\frac{F_i(\lceil k\rceil S_i)}{kF_i\left(S_i\right)}\min_{I\ge 0}\left(\frac{m\cdot(1-C_s'/P_s)+k\cdot(1-2C_s'/D_{\min})}{m+1}\right). \qquad \text{(D-27)}$$

An equivalent form of is

$$Z(i) \ge \frac{1-n\alpha}{1/\gamma'+1}\frac{F_i(\lceil k\rceil S_i)}{kF_i\left(S_i\right)}\min_{I\ge 0}\left((1-C_s'/P_{\min})+\frac{k(1-2C_s'/D_{\min})-(1-C_s'/P_s)}{m+1}\right) \,. \qquad \text{(D-28)}$$

It can be verified that the right hand side of (D-28) is minimized either at m=0 or $m=\infty$, so

$$Z(i) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\frac{F_i(\lceil k\rceil S_i)}{kF_i\left(S_i\right)}\min\left((1-C_s'/P_s),\ k\cdot(1-2C_s'/D_{\min})\right) \,. \qquad \text{(D-29)}$$

By (3.6) and (3.7), we have

$$F_i(\lceil k\rceil S_i)/F_i(S_i) = \lceil k\rceil/\mu_i \ge \lceil k\rceil/\mu \,. \qquad \text{(D-30)}$$

By substituting (D-30) into (D-29), we get

$$Z(i) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\frac{\lceil k\rceil}{k\mu}\min\left((1-C_s'/P_s),\ k\cdot(1-2C_s'/D_{\min})\right) \,. \qquad \text{(D-31)}$$

By substituting (A-27) into (A-10), we have

$$W(1,\ \Gamma) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\frac{\lceil k\rceil}{k\mu}\cdot\min\left((1-C_s'/P_s),\ k(1-2C_s'/D_{\min})\right) \,. \qquad \text{(D-32)}$$

By (3.6), we know that

$$W\left(1,\ \Gamma\right) = \sum_{i=1}^{n} F_i(D_i)/D_i = \lceil k\rceil/k\cdot W\left(\lceil k\rceil/k,\ \Gamma\right) \,. \qquad \text{(D-33)}$$

By substituting (D-33) into (D-32) and rearrange the resulting inequality, we have

$$W\left(\lceil k\rceil/k,\ \Gamma\right) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\frac{1}{\mu}\min\left((1-C_s'/P_s),\ k\cdot(1-2C_s'/D_{\min})\right) \,. \qquad \text{(D-34)}$$

Since (D-34) is true $\forall\Gamma\in\Omega^*$, we know that

$$\inf_{\Gamma\in\Omega^*}\left(W(\lceil k\rceil/k,\ \Gamma)\right) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\frac{1}{\mu}\min\left((1-C_s'/P_s),\ k(1-2C_s'/D_{\min})\right) \,. \qquad \text{(D-35)}$$

Then, by substituting (D-35) into (D-5), we get

$$W^*(\lceil k\rceil/k) \ge \left(1-n\alpha\right)\cdot\frac{1}{1/\gamma'+1}\frac{1}{\mu}\min\left((1-C_s'/P_s),\ k\cdot(1-2C_s'/D_{\min})\right) \,. \qquad \text{(D-36)}$$

Comparing (D-36) with (D-1), we have proven the theorem.

Q.E.D.