GENERAL SCHEDULABILITY BOUND ANALYSIS

AND ITS APPLICATIONS IN REAL-TIME SYSTEMS

A Dissertation

by

JIANJIA WU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2006

Major Subject: Computer Science

GENERAL SCHEDULABILITY BOUND ANALYSIS

AND ITS APPLICATIONS IN REAL-TIME SYSTEMS

A Dissertation

by

JIANJIA WU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Wei Zhao
	Douglas K. Loh
Committee Members,	Jyh-Charn Liu
	Rabi N. Mahapatra
Head of Department,	Valerie E. Taylor

May 2006

Major Subject: Computer Science

ABSTRACT

General Schedulability Bound Analysis and Its Applications in Real-time Systems.

(May 2006)

Jianjia Wu, B.S., Tianjin University;

M.S., Beijing University of Aeronautics and Astronautics

Co-Chairs of Advisory Committee: Dr. Wei Zhao
Dr. Douglas Loh

Real-time system refers to the computing, communication, and information system with deadline requirements. To meet these deadline requirements, most systems use a mechanism known as the schedulability test which determines whether each of the admitted tasks can meet its deadline. A new task will not be admitted unless it passes the schedulability test. Schedulability tests can be either direct or indirect. The utilization based schedulability test is the most common schedulability test approach, in which a task can be admitted only if the total system utilization is lower than a pre-derived bound. While the utilization bound based schedulability test is simple and effective, it is often difficult to derive the bound. For its analytical complexity, utilization bound results are usually obtained on a case-by-case basis. In this dissertation, we develop a general framework that allows effective derivation of schedulability bounds for different workload patterns and schedulers. We introduce an analytical model that is capable of describing a wide range of tasks' and schedulers' behaviors. We propose a new definition of utilization, called workload rate. While similar to utilization, workload rate

enables flexible representation of different scheduling and workload scenarios and leads to uniform proof of schedulability bounds. We introduce two types of workload constraint functions, s-shaped and r-shaped, for flexible and accurate characterization of the task workloads. We derive parameterized schedulability bounds for arbitrary static priority schedulers, weighted round robin schedulers, and timed token ring schedulers. Existing utilization bounds for these schedulers are obtained from the closed-form formula by direct assignment of proper parameters. Some of these results are applied to a cluster computing environment. The results developed in this dissertation will help future schedulability bound analysis by supplying a unified modeling framework and will ease the implementation practical real-time systems by providing a set of ready to use bound results.

DEDICATION

To my wife and parents

ACKNOWLEDGEMENTS

This dissertation would not have been possible to complete without the help of so many great people.

I would like to thank Professor Wei Zhao, my supervisor, for his constant guidance, support, inspiration, and encouragement through all my Ph.D. study. He taught me the principles of doing research and ways to approach research problems. He showed me how to ask questions, how to express my ideas, and how to write research papers that can be understood by others. He gave me insightful advice for my research as well as daily life. He is a master teacher and his comments always hit the mark. What I learned from him will benefit my future.

Special thanks go to Professor Douglas Loh, my co-supervisor, who introduced me to the Texas A&M University and provided me continuous support through the past years. He taught me how to broaden my mind, not narrow it. He advised me on how to think systematically and how to view things from different perspectives. Most importantly, he showed me how to respect others. He is an outstanding supervisor, and a great friend.

I want to express my sincere gratitude to Professor Jyh-Charn Liu for his patient, support, encouragement and trust. He was always there to discuss with me about my research problems and he provided many valuable comments. He spent hours and hours of time helping me revise my research papers and showing me why the revisions are better. He is one of those exceptional professors who do not show you the solution directly, but guide you through the process of reaching the solution.

Many thanks go to Professor Rabi Mahapatra for serving on my committee and for sharing his research experiences with me.

I want to thank Professor Jerry Stuth for allowing me to complete my Ph.D. study while working full time for him.

I want to thank all fellow students in my research group, especially Zhibin Mai, Dan Cheng, Shengquang Wang, and Nan Zhang, for their collaboration and insightful comments.

Finally, I would like to thank my parents, sister, and brother for their unconditional help and support. I owe a lot to my beloved wife who encouraged me when I was depressed, listened to me when I was complaining, and believed in me even when I doubted myself. This dissertation could not have been completed without my lovely daughter who is my source of inspiration.

NOMENCLATURE

α	Protocol overhead ratio for weighted round robin and timed token ring schedulers, $\alpha = \tau / TTRT$
C^j	The increment of an s-shaped workload constraint function at the j^{th} period, $j = 1, 2, \ldots, L$
C	The constant increment of an s-shaped workload constraint function after the L^{th} period
d	The worst case (i.e., largest) delay of all jobs in a task
Δ	A positive integer defined by the relationship of $k = \Delta\lambda$
D	Relative deadline of a task
η	Heterogeneity of a task set, $\eta = V(\lceil k/\lambda \rceil, \Gamma)$
$f(t)$	Workload function
$F(I)$	Workload constraint function
$g(t)$	Service function
$G(I)$	Service constraint function
Γ	Task set
γ	Normalized token rotation frequency for weighted round robin and timed token right schedulers, $\gamma = \lfloor D_{\min} / TTRT \rfloor$ where D_{min} is the shortest relative deadline of all tasks
$V(\hbar, \Gamma)$	Heterogeneity function of task set Γ
k	Normalized deadline of a task, $k = D/P$
L	Number of $C^j s$ in an s-shaped workload constraint function
λ	Degree of deadline inversion
P	Period of a periodic task
μ	Task set workload burstness of s-shaped tasks
S	Segement length parameter in an s-shaped workload constraint function
τ	Token rotation overhead which is the time spent in token rotation per round

t_d	Job absolute deadline
t_r	Job release time
t_f	Job completion time
T	A task
TRT	Token rotation time which is time to finish last token rotation
TTRT	Target token rotation time for weighted round robin and timed token ring schedulers
θ	Scaling parameter used for workload rate measurement
$W(\theta, \Gamma)$	Workload rate of task set Γ
$W^*(\theta)$	Schedulability bound

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

A. Real-time Systems

Real-time system refers to the computing, communication, and information system with deadline requirements. Real-time systems and can be divided into soft real-time systems and hard real-time systems. In a soft real-time system, violation of the deadline requirement may lead to some level of performance suffering or economic loss, while in hard real-time systems, catastrophe results will happen. Examples of soft real-time systems include video conferencing, voice over IP, and many others. In a video conferencing/voice over IP system, the video/voice data packages need to be delivered from one end to the other in a given time interval. Violation of the deadline requirements affect the smoothness of the conference or conversation, but typically will not result in catastrophe outcomes. Examples of hard real-time systems include missile control, radar monitoring, aircraft traffic control, and many others, in which missing deadline requirements may lead to huge economic or human life loss, e.g. failure of delivery of routing commands to an aircraft before deadline may result in aircraft collision.

A typical real-time system includes three major components: resource, task, and scheduler. A task is a sequence of jobs that together accomplish certain mission, and the resource is what needs to be consumed in order to finish a job. Since the resource in a

---

This dissertation follows the style of *IEEE Transactions on Computers.*

system is shared, different jobs may want to use the resource at the same time and thus confliction will happen. The scheduler is the unit that decides which job should use the resource in case of such resource usage confliction. In a voice over IP system, a conversation can be treated as a task and delivering of each voice package is a job. The network bandwidth is the resource to be consumed for the delivery the voice packages. The routers responsible for sending/receiving data packages can be considered as the schedulers.

B. Schedulability Test

To meet the deadline requirements, most real-time systems use a mechanism known as the schedulability test (also called admission control), which determines whether each of the admitted tasks can meet its deadline. A new task will not be admitted unless it passes the schedulability test. An admitted task will be guaranteed to meet its deadlines throughout its mission.

The schedulability test can be either direct or indirect. Direct schedulability test explicitly calculates the worst-case delays (the length of the time interval from the instant a job request the resource to the instant the job is finished) of the tasks in order to determine the permissibility of a new task. This type of test is more accurate, but usually has high run-time computing cost in calculating the delays.

An indirect schedulability test does not compute the delays, but tests selected system parameters to determine the schedulability of a new task. The utilization based schedulability test is the most common approach, in which a task can be admitted only if the total system utilization is lower than a pre-derived bound. The major advantages of this schedulability test are as follows:

1) It is very efficient with a complexity of O(1). Unlike the direct schedulability test that need to evaluate the schedulability of each task in the system (and the new one) upon the arrival a new task, the utilization based schedulability test only needs to check whether the total system utilization (including the new task) is lower than the pre-derived utilization bound.
2) It provides an operation margin for system administrators and thus improves the stability of a real-time system. During design phase, the system designer can set an upper bound of utilization that is lower than the pre-derived bound. By providing such safety margin, the system can work smoothly even some tasks accidentally violate their load constraints, or when some system parameter changes, e.g. clock skew.

Various utilization bounds have been derived in the literature, and many have been applied to implementation of mission critical applications. Some of the most notable results include 69% (and extensions) for the Rate/Deadline Monotonic scheduler (RMS/DMS) [34], [43], [49], [56], [57], and [66]; 100% for Earliest Deadline First scheduler (EDF) [49]; and 33% for the Timed Token protocol (TTP) [5], [52], [75], and [80]; etc. Utilization bounds of RMS/DMS in multiprocessor systems have also been derived in [9], [10-12], [29], and [62]. Some important utilization bounds for non-periodic systems are derived in [1-3], and [74]. A more detailed summary of the literature is given in the related work section.

C. Problems

Though many bound results have been obtained for existing scheduling systems, most of the bounds are for periodic tasks, and it is difficult to generalize the utilization bound method to non-periodic system due to the following two problems:

1) Ambiguity in defining utilization for non-periodic tasks. Utilization is a measurement of the resource consumption rate within a certain time window (referred to as a measuring window). Typically, for periodic systems, task periods have been used as the measuring window. It is difficult to extend this definition to the domain of non-periodic tasks because one cannot have a well-defined notion of "period". One certainly could define a long-term stable utilization with the measuring window length being infinitely large, but this type of definition may not correctly reflect the resource demand within the deadline. Because of this, in [1] and [9], the authors proposed to define the utilization by setting the length of the measuring window as the relative deadline of the task. Though works in some cases, this definition used a fixed interval (the relative deadline of the task) to measure the workload and we noticed that deriving the bound result using a fixed interval is very difficult in some scheduling systems. To derive utilization bounds, we must have a flexible, robust notion of utilization that can be applied to a broad range of workloads and schedulers. The definition should correctly reflect the resource demand and facilitate derivation of the bounds.

2) Ad hoc-ness in the derivation of the utilization bound. Most utilization bound results are obtained case-by-case, and the method developed for one system cannot be easily applied to another. The difficulty of the bound derivation can be attributed to the high complexity of the underlying optimization problem since one must find an optimal (lower) bound of the utilization in an infinite space of non-schedulable task sets, but more importantly, to the lack of general system model and bound derivation methodology.

D. Related Work

In their seminal work [49], the authors derived the well-known 69% utilization bound for the RMS on single processor systems, where relative deadlines of periodic tasks are equal to their periods. A rich collection of utilization bounds have been derived since then for different systems. This result has been extended to arbitrary deadline assignment schemes in [42], [43] and [66]. In [40] and [46], the authors improved the bound by exploiting the ratio between the longest and shortest task periods. The work in [22] and [39] further improved the bound result with the concept of the harmonic chain that exploits the divisibility between periods. The authors in [34] introduced an algorithm that transforms the periodic task into a harmonic task set, which has a workload bound of 1, and proved that the algorithm performs better than the bound derived in [46] and [49], with the cost of higher complexity.

Utilization bounds of static priority schedulers on the time token protocol in FDDI networks were derived in [5], [52], [75], and [80]. The utilization bound for static priority schedulers in a network environment have been studied in [74]. Utilization bounds for non-periodic tasks have been addressed in [1-3] and [51]. Utilization bounds for RMS/DMS in multi-processor systems have been studied in [9-11], [29], and [51]. Schedulability analysis for weighted round robin schedulers has been conducted in [8].

Generalizing the definition of utilization from periodic tasks to aperiodic tasks has been studied in [1-3], [9], [56], and [74]. In deriving the utilization bound for RMS with multiframe and general real-time task models, the authors in [56] and [57] proposed a maximum average utilization that allows calculation of utilization in an infinite measuring window. In the analysis of the utilization bound in a multi-node network environment with leaky bucket packet sources, the authors in [74] used a utilization definition that is based on the sustainable rate in the leaky bucket function. To derive the utilization bound for non-periodic tasks and multiprocessor systems, the authors in [1-3] and [9] proposed a utilization definition that is based on relative deadlines of tasks, instead of periods.

The linear programming method that has been proposed for finding utilization bounds when task parameters are known a priori has been studied in [22], [41], and [65]. The work in [13] and [14] introduced a new schedulability test which is similar to utilization based admission control. Specifically they proved that for periodic tasks with RMS, a task set is schedulable if $\Pi(u_i + 1) \leq 2$, where u_i is the utilization of the i-th task. Some non utilization based schedulability bound analysis for static priority scheudlers are done in [44] and [47].

Generalization of a periodic task model was proposed in [52] and [56], in which the authors derived a bound result for multiframe tasks which allows jobs in the same task to have different size, provided that the relative deadline is same as period length.

Using workload constraint function to model tasks can be traced back to [23], [24], i.e. leaky bucket constraints of network traffic. This concept was expanded in [74] to analyze the utilization bounds of static priority schedulers. The general model for real-time tasks proposed in [56] shares a similar concept, and it corresponds to a special group of workload constraint functions in multiframe forms. The idea of modeling schedulers with service constraint functions originated in [16], [18], [19], [25], and [64]. Workload constraint and service constraint functions have been used for direct schedulability test in [6], [7], [15], [25], [26], [47], [67], and [68] among many others, but none of them is used for utilization based test.

E. Dissertation Contributions

In this dissertation, we introduce an analytical model that is capable of describing a wide range of tasks and schedulers' behaviors[1]. In addition to address the problems mentioned above, we broaden the utilization bound derivation techniques using the following approach:

1) We propose a new definition of utilization, called workload rate, which measures the resource demand within a time window of length proportional to

---

[1] Some of the results of this dissertation have been published in [73].

the deadline of a task, so that it can be used to characterize both periodic and non-periodic tasks in the same framework. Taking the relative deadline of the task length of the measuring window to define the utilization was first proposed in [1] and [9]. Several other key system parameters are characterized with respect to the workload rate in formulation of the utilization bound solutions.

2) We introduced two special types of task workload models, i.e. the s-shaped and r-shaped. The s-shaped workload model is an extension of the classical periodic task model and allows more accurate and flexible characterization of task workload requirements. The r-shaped workload model is an extension of the classical leak-bucket task model widely used in network environment.

3) On the basis of the network calculus framework [15], [16], [18], [19], and [23-25], a new bound derivation methodology is proposed. We derive some key relationships between workload and services, to arrive at a lower bound of workload rate for arbitrary services and schedulers. In previous work, the search for utilization bound was usually made along the boundary between the spaces of schedulable and non-schedulable task sets. Knowing that finding the boundary of the two spaces is already a major undertaking, we directly derived the schedulability bound by solving a minimization problem over the entire task set population. As a result, the utilization results (of schedulability testing) are applicable to a much broader range of task models and schedulers.

4) To illustrate the effectiveness of our new methodology, we explicitly derive a set of parameterized workload rate bound for static priority schedulers, weighted round robin schedulers, and timed token ring schedulers.

   - A closed-form bound formula is obtained for static priority schedulers. The bound is parameterized for different priority assignments and for various task releasing patterns. We show that when the parameters are set properly, existing bounds can be easily obtained from our generalized bound formula, including: the system has periodic tasks whose deadlines are equal to periods. Tasks are scheduled by a rate monotonic scheduler [49]; the system has periodic tasks whose deadlines are less than periods. Tasks are scheduled by a rate monotonic scheduler [43] and [66]; the system has periodic tasks whose deadlines are multiples of periods. Tasks are scheduled by a rate monotonic scheduler [34]; the system has multiframe tasks whose deadlines are equal to periods. Tasks are scheduled by a rate monotonic scheduler [57]. To our knowledge, no current literature covers as a wide range of systems as our methodology.
   - A set of closed-form bound formulae are obtained for weighted round robin schedulers with different weight assignment schema.
   - A set of closed-form bound formulae are obtained for TTP schedulers with different weight assignment schema. We shown that existing bounds are special cases of our newly derived bound and can be obtained from the new bound, including the 33% utilization bound for the TTP with periodic tasks

and normalized deadline assignment scheme [5], the utilization bound for TTP scheduler with periodic tasks and optimized weight assignment [55].

To best of our knowledge, no current literature covers as a wide range of systems as our methodology.

F. Dissertation Outline

This dissertation is organized as follows. Chapter II introduces the task, scheduler and schedulability bound models. In Chapter III, schedulability bounds for static priority are analyzed. Chapter V derives closed-form bounds for weighted round robin schedulers with different weight assignment schema. Chapter VI derives closed-form bounds for timed token ring schedulers. In Chapter VII, application of the bound results to a cluster environment is discussed. Summary and conclusions are given in Chapter VIII.

CHAPTER II

SYSTEM MODEL

A. Task, Task Set, Workload, and Service Functions

We assume that a single processor computing system is to serve a task set $\Gamma = \{T_1,\ T_2,\ ...,\ T_n\}$, where T_i is the i-th task. When the context is clear, we may omit index i in the subsequent discussions. Each task is composed of a sequence of jobs. The worst-case execution time of a job is called the job size, which is measured in second. 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 of the job, and the worst-case (i.e., largest) delay of all jobs in a task is denoted by d^*. Within a task, the jobs have the same relative deadline, but may not necessarily have the same size. Jobs within a task are executed in a first come, first served order.

To characterize the resource demand of task T analytically, we define $f(t)$, the workload function for T, as follows:

$$f(t) = \text{the summation of the sizes of all the jobs from } T \text{ released in } [0,\ t]. \quad \text{(II-1)}$$

Figure 1 is an example of the workload function. Figure 1.a illustrates the job sizes and arrival time instants. As can be seen, there is a job of size 5 released at time 1, followed by a job of size 2 at time 5, a job of size 2 at time 9, and so on. Figure 1.b draws the workload function. It is clear that there is a jump of the size of the job size at the release instant of each job.

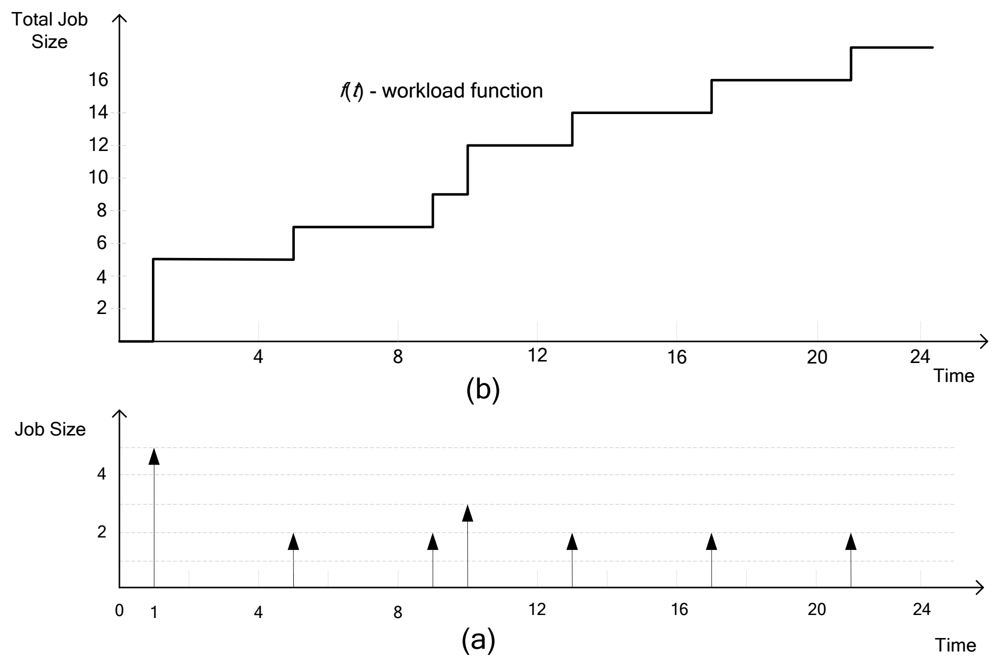

Figure 1. Example of Workload Function.

We say $f(t)$ is a periodic workload function if it is of the following form:

$$f(t) = \left\lceil \frac{t-\phi}{P} \right\rceil C\,, \qquad \text{(II-2)}$$

where C is the job size, ϕ is the phase, and P is the period of the task. It is easy to see that a periodic task release its first job of size C right after time ϕ and releases a new job same size every P time unit. We say a task is a periodic task if its workload function is periodic. Periodic task exists in many real-time systems, e.g. periodic sampling task in a digital sampling system, periodic target tracking task in a radar monitoring system. Figure 2 is an example periodic task with period of 7, job size of 4, and phase being 1.

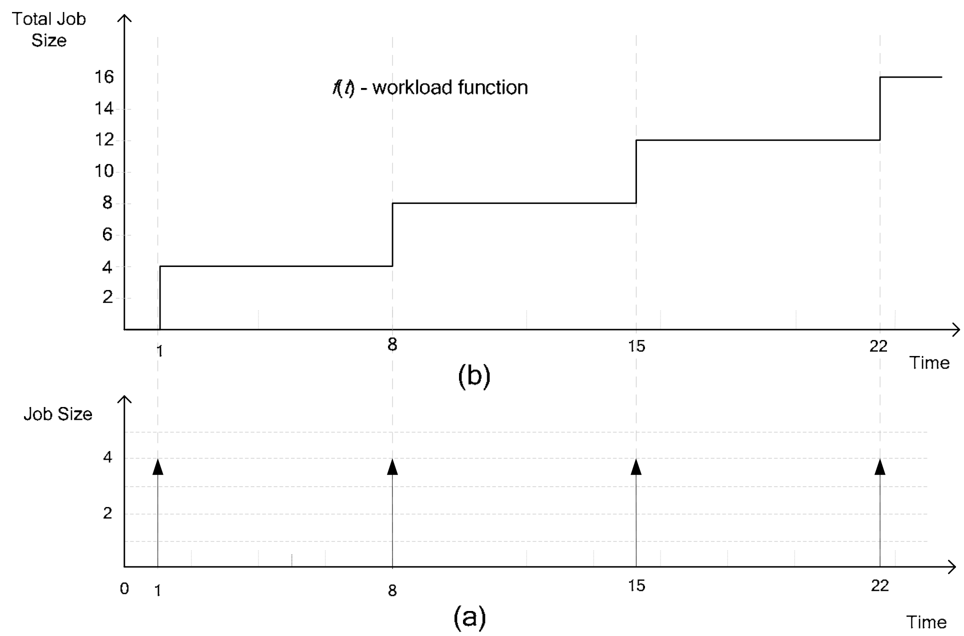

Figure 2. An Example Periodic Task.

Similarly, to characterize the actual processor time received by task T, we define $g(t)$, the service function for T, as follows:

$$g(t) = \text{the total execution time rendered to jobs of task } T \text{ during } [0,\ t]. \quad \text{(II-3)}$$

Let us consider a real-time systems with two periodic tasks $\Gamma = \{T_1,\ T_2\}$ where $P_1 = 5$, $C_1 = 2$, $D_1 = 3$, and $P_2 = 6$, $C_2 = 1$, $D_2 = 4$. We assume the jobs are served in a first-come-first-serve order. We use $J_{i,k}$ to denote the k^{th} job from task T_i. Figure 3 illustrates the arrival time of the jobs (Figure 3.a and 3.b), the CPU execution sequence (Figure 3.c), and the workload and service functions for the two tasks (Figure 3.d).

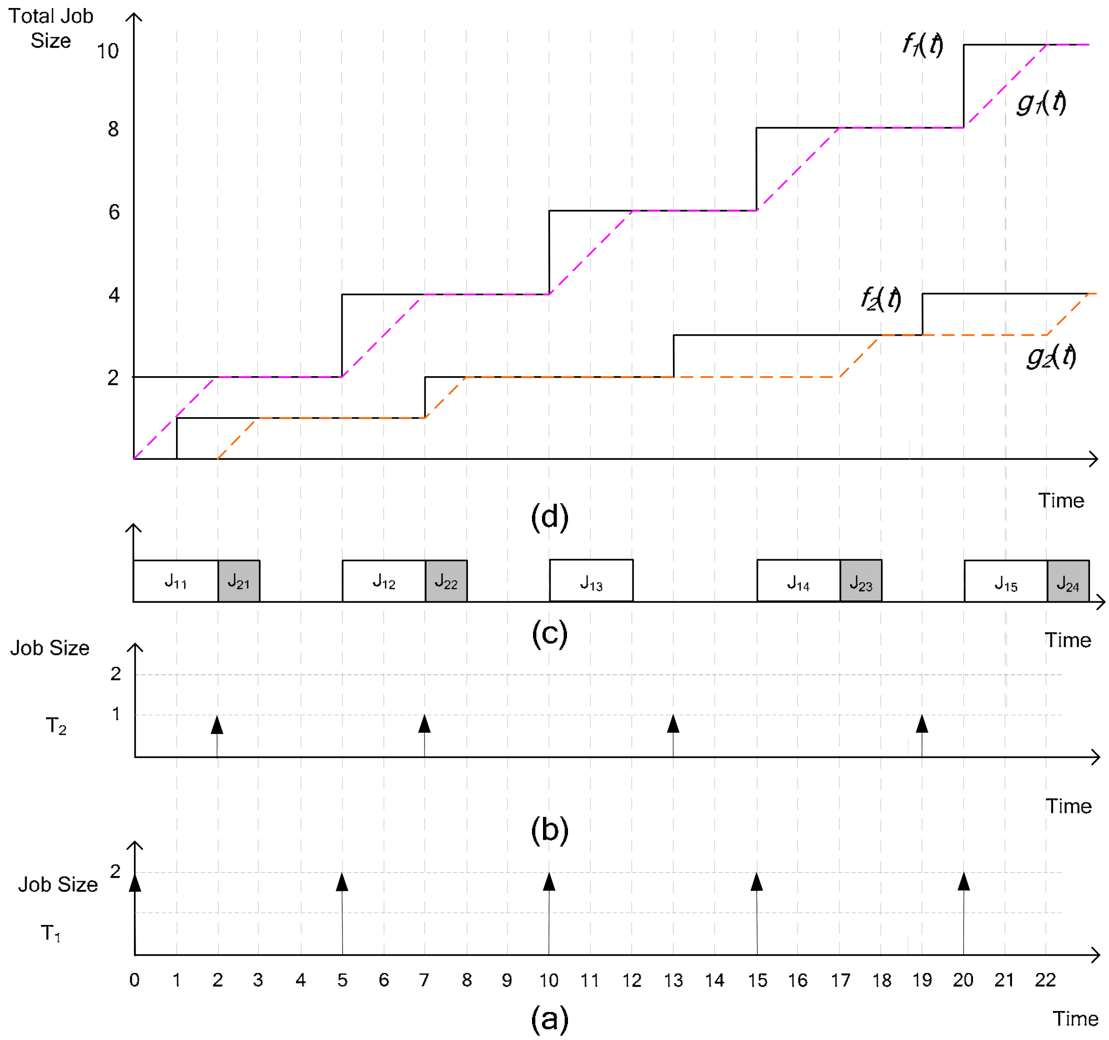

Figure 3. An Example Task Set, Their Execution Sequence, Workload and Service Functions.

Based on the definitions of d^*, $f(t)$ and $g(t)$, it can be verified that

$$d^* = \sup_{t \geq 0} \left(\inf \left(\tau \mid f(t) \leq g(t+\tau) \right) \right). \tag{II-4}$$

That is, the worst case delay of a task is the maximum horizontal distance between its workload and service function. Interested reader can check Figure 3 and find out that the worst case delay for task T_1 is 2 and 5 for task T_2.

In a real-time system, a major goal of the schedulability test algorithm is to check the truthfulness of

$$d^* \leq D. \tag{II-5}$$

One may want to use (II-4) to calculate d^* and then compare the result with D to test the schedulability. However, this method may not be suitable for online operation because the exact forms of $f(t)$ and $g(t)$ may not be available when schedulability test is made. For example, in an online real-time conference system, the exact workload of the video traffic will be unknown until the conference is finished since the sizes of video frames depend on the movement of the participants during the meeting and other dynamic factors. Furthermore, even if $f(t)$ and $g(t)$ are available, e.g. an online play-back movie, they are often too cumbersome to handle. A practical solution is using some alternative simple forms of $f(t)$ and $g(t)$ that can be obtained during schedulability test.

B. Workload Constraint Functions

Much work has been done on the alternatives of $f(t)$. For example, in periodic task model, a typical alternative of $f(t)$ is

$$F(t) = \lceil t / P \rceil C , \tag{II-6}$$

where C is the maximum job size and P is the minimum inter-job separation time. Though this alternative maybe accurate for periodic tasks, it can over-estimate the resource demand for non-periodic tasks [57], and would lead to pessimistic schedulability decisions. Let us consider the example in Figure 1 and model the task workload use the function defined in (II-6). Since the maximum job size of the task is 5 and the minimum length of job separation time is 1, we have $C = 5$ and $P = 1$. As a result, we have the function

$$F(t) = 5\lceil t \rceil . \tag{II-7}$$

Figure 4 compares this alternative with the original f function. One can notice degree of over-estimation.

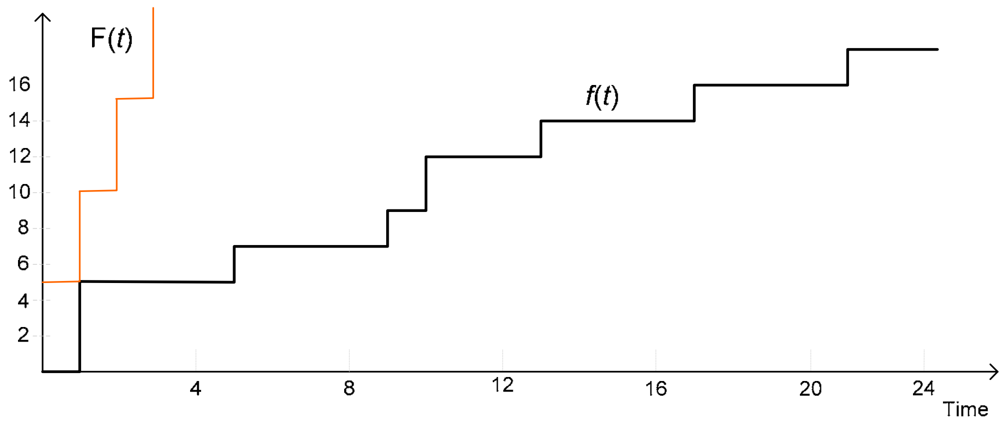

Figure 4. Over-estimation Problem of the Periodic Task Model.

A different alternative of $f(t)$ is the workload constraint function $F(I)$ that satisfies that for any $0 \le I \le t$

$$f(t) - f(t-I) \le F(I). \quad \text{(II-8)}$$

The $F(I)$ in form of (II-8) was first introduced in [23] and [24] and has been widely used [15], [18], [19], [45], and [74]. By convention, $F(0) = 0$ and $F(I)$ is non-decreasing. $F(I)$ is an upper bound of total size of jobs can be released in any time window $[t-I,\ t]$. We use I in (II-8) because F defined on the domain of time intervals, while $f(t)$ is defined in the domain of time. Typically, a workload constraint function should have the following properties as discussed in [16].

Property 1 (non decreasing): For $I \ge 0$ and $\Delta \ge 0$,

$$F_i(I + \Delta) \ge F_i(I). \quad \text{(II-9)}$$

Property 2 (triangle relationship): For $I_1 \geq 0$ and $I_2 \geq 0$,

$$F_i(I_1 + I_2) \leq F_i(I_1) + F_i(I_2). \quad \text{(II-10)}$$

The workload constraint function defined in (II-8) is tighter than the alternative function defined in (II-6), since for systems with periodic task model, it can be verified that $F(I) = \lceil I / P \rceil C$ is a workload constraint function satisfies (II-8). Figure 5 illustrates a workload constraint function for the task introduced in Figure 3. By checking the arrival pattern of the jobs from the task, we know that in any interval of length 1, the total job size is no more than 5 units, 7 in any interval of length 5, and 9 in any interval of length 8, etc… Thus, we can verify that the $F(I)$ satisfies the constraint in (II-8). One can notice that this alternative function $F(I)$ is tighter than the one used in Figure 4. In this dissertation we will use $F(I)$ defined in (II-8) as the chosen alternative of $f(t)$.

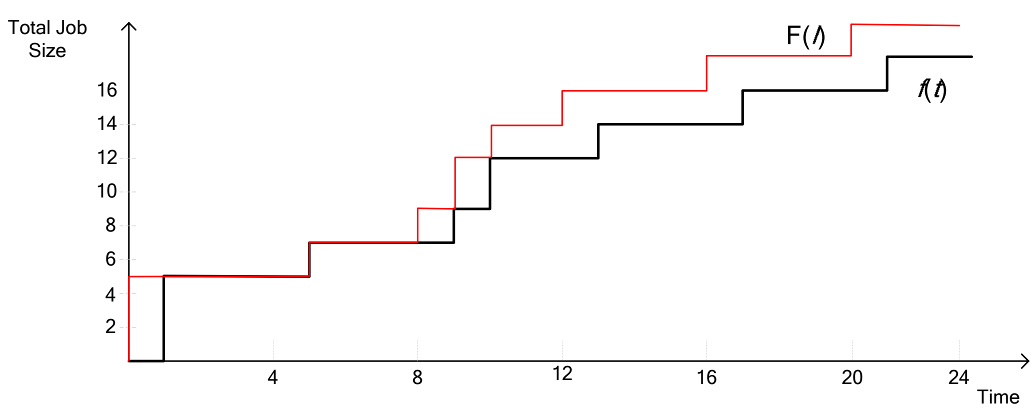

Figure 5. Modeling Task Workload with Workload Constraint Function.

In most practical cases, the F function can be obtained based on the known workload properties of the tasks. For example, in a network environment, the incoming traffic is typically regulated by a leaky bucket controller which controls that in any interval of length I, the total packages went through the controller is no more than $\sigma + \rho I$. Thus $F(I) = \sigma + \rho I$ is a workload constraint function satisfies (II-8).

1. S-shaped Workload Constraints

Knowing that workload constraint functions can exist in many different forms and deriving schedulability bound for arbitrary workload constraint function may be very challenging, we start with a special workload constraint function, namely the staircase-shaped (s-shaped) workload constraint function.

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. We assume that the segment length S is fixed, and the increments may not be identical for the first L segments where L is a parameter in the function.

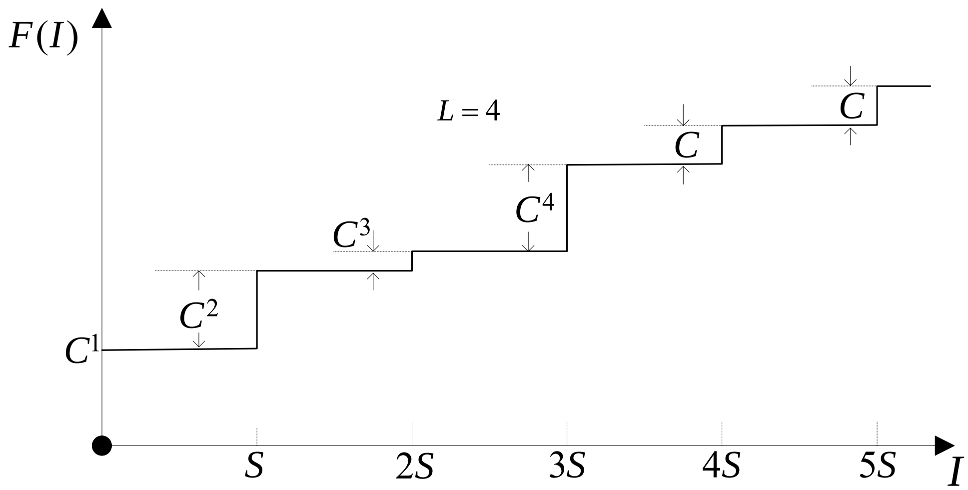

Figure 6. An Example S-shaped Workload Constraint Function.

Formally, an s-shaped workload constraint function can be expressed as follows:

$$F(I)=\begin{cases}\sum_{j=1}^{a}C^{j} & a\le L\\ \sum_{j=1}^{L}C^{j}+(u-L)C & a>L\end{cases} \qquad \text{(II-11)}$$

where $a=\lceil I/S\rceil$. C^j is the increment at the beginning of the j^{th} segment, and C is the constant increment after the L^{th} segment. Figure 6 shows an example of the s-shaped workload constraint function. When $L=\infty$, an s-shaped constraint function reduces to the general real-time task model, as the one defined in [56].

We say that an s-shaped function $F(I)$ is smooth when

$$C^{1}\ge C^{2}\ge\cdots\ge C^{L}\ge C. \qquad \text{(II-12)}$$

That is, if an s-shaped constraint function is smooth, then its increments over time are monotonically non-increasing. The smoothness property greatly simplifies the schedulability analysis.

Any non-smooth s-shaped workload constraint function can be converted into a smooth one. One simple algorithm is:

Step 1. Locate the first C_j in the function such that $C_j < C_{j+1}$;

Step 2. Replace both C_j and C_{j+1} with $(C_j + C_{j+1})/2$;

Step 3. Repeat Steps 1 and 2 until no such C_j exists;

Step 4. Replace the C_j s that are less than C with C.

It can be easily verified that the result of the above process will produce a constraint function that is s-shaped and still meets the definition given in (II-8). For example, consider the non-smooth s-shaped constraint function in Figure 6 with segment length $S = 8$, $C_1 = 5$, $C_2 = 5$, $C_3 = 1$, $C_4 = 5$, and $C = 2$. This function can be transformed to a smooth one with $S = 8$, $C_1 = 5$, $C_2 = 5$, $C_3 = 3$, $C_4 = 3$, and $C = 3$.

In the rest of this dissertation, unless otherwise specified, we only consider smooth s-shaped workload constraint functions. The s-shaped workload constraint functions cover a broad range of tasks and have analytical properties that facilitate the derivation of workload rate bounds. The following example illustrates how to model the multiframe tasks with s-shaped workload constraint function.

Example. S-shaped workload constraint functions for multiframe tasks. As defined in [31], a multiframe task is expressed in the form of $\left(\left(E_i^0,\ E_i^1,\ ...,\ E_i^{N-1}\right),\ P_i\right)$ where P_i is the minimum job separation time and the execution time of the j^{th} job is $E_i^{(j-1) \bmod N}$. For instance, ((3, 1), 3) denotes a task whose minimum job separation time is 3, and its execution time alternates between 3 and 1. A multiframe task is said to be Accumulatively Monotonic (AM) [57], if the total execution time for the first *j* frames is the largest among all size *j* frame sequences for all *j*, *j* = 1, 2, ….

Recall that an s-shaped constraint function is characterized by parameters S, L, C^1, C^2, …, C^L, and *C*. Given a multiframe task $\left(\left(E_i^0,\ E_i^1,\ ...,\ E_i^{N-1}\right),\ P_i\right)$, we can construct its corresponding $F(I)$ by assigning the following parameter values: $S = P_i$, $L = \infty$, $C^1 = \phi^1$, and $C^j = \phi^j - \phi^{j-1}$, $j = 2,\ 3,\ ...$, where ϕ^j, $j = 1,\ 2,\ ...$, is defined as

$$\phi^j = \max_{\ell=0,\ 1,\ 2,\ ...}\left(f\left((\ell + j)P\right) - f\left(\ell P\right)\right). \tag{II-13}$$

It is can be verified that for $t \geq 0$, $I \geq 0$, the following inequality holds for $F(I)$ just constructed,

$$f(t+I) - f(t) \leq F(I), \tag{II-14}$$

and it is a valid workload constraint function. By definition, when the multiframe task is AM we know that $\phi^1 = E^0$ and $\phi^2 = E^0 + E^0$. And thus, we have $C^1 = E^0$ and $C^1 = E^1$. Note that the newly constructed s-shaped function is not necessarily smooth.

Note that the notion of workload and service constraint function defined in (II-8) and (II-41) are not entirely new. Similar definitions have been proposed in the literature, e.g., the burst-ness constraint function in [23], the arrival curve and service curves in [15] and

[16], the rate controlling function in [46], and the workload constraint functions in [45], [72], and [74], just to name a few. But, little, if any, of its effect has been explored for schedulability bound analysis.

For any s-shaped workload constraint function, we have the following property.

Lemma 2-1. For an s-shaped constraint function F and two positive integers ℓ and ℓ', if $\ell \le \ell'$, then

$$\frac{F(\ell S)}{\ell} \ge \frac{F(\ell' S)}{\ell'} \quad . \tag{II-15}$$

Proof. We prove this lemma by induction. It is trivial that (II-15) holds for $\ell = \ell'$. Assume that (II-15) holds for $\ell' = \ell + m$. That is,

$$\frac{F(\ell S)}{\ell} \ge \frac{F((\ell + m)S)}{\ell + m} \quad . \tag{II-16}$$

Now we will prove (II-15) is true for $\ell' = \ell + m + 1$. That is,

$$\frac{F(\ell P)}{\ell} \ge \frac{F((\ell + m + 1)S)}{\ell + m + 1} \quad . \tag{II-17}$$

To establish the lemma, we only need to show

$$\frac{F((\ell + m)S)}{\ell + m} \ge \frac{F((\ell + m + 1)S)}{\ell + m + 1} \quad . \tag{II-18}$$

We will prove (II-18) in two cases.

Case 1: $\ell + m < L$. By (II-11), we have

$$F((\ell+m+1)S) = F((\ell+m)S) + C^{\ell+m+1}. \quad \text{(II-19)}$$

By multiplying $(\ell+m)$ on both sides of (II-19), we get

$$(\ell+m)F((\ell+m+1)S) = (\ell+m)F((\ell+m)S) + (\ell+m)C^{\ell+m+1}. \quad \text{(II-20)}$$

By (II-12), we know that $C^{\ell+m+1} \le C^j$, for all $j \le \ell+m+1$, and thus,

$$(\ell+m)C^{\ell+m+1} \le \sum_{j=1}^{\ell+m} C^j. \quad \text{(II-21)}$$

Substituting (II-21) into (II-20), we have

$$(\ell+m)F((\ell+m+1)S) \le (\ell+m)F((\ell+m)S) + \sum_{j=1}^{\ell+m} C^j. \quad \text{(II-22)}$$

By (II-11),

$$F((\ell+m)S) = \sum_{j=1}^{\ell+m} C^j. \quad \text{(II-23)}$$

Then substituting (II-23) into (II-22), we have

$$(\ell+m)F((\ell+m+1)S) \le (\ell+m+1)F((\ell+m)S). \quad \text{(II-24)}$$

(II-24) is equivalent to (II-18).

Case 2: $\ell+m \ge L$. By (II-11), we have,

$$F((\ell+m+1)S) = F((\ell+m)S) + C. \quad \text{(II-25)}$$

By multiplying $(\ell+m)$ on both sides of (II-25), we get

$$(\ell + m)F((\ell + m + 1)S) = (\ell + m)F((\ell + m)S) + (\ell + m)C . \tag{II-26}$$

By (II-12), $C \le C^j$, for all $j \le \ell + m + 1$. Thus,

$$(\ell + m)C \le \sum_{j=1}^{L} C^j + (\ell + m - L)C . \tag{II-27}$$

By (II-11)

$$F((\ell + m)S) = \sum_{j=1}^{L} C^j + (\ell + m - L)C . \tag{II-28}$$

By substituting (II-28) into (II-27), we get

$$(\ell + m)C \le F((\ell + m)S) . \tag{II-29}$$

Substituting (II-29) into (II-26), we have

$$(\ell + m)F((\ell + m + 1)S) \le (\ell + m + 1)F((\ell + m)S) . \tag{II-30}$$

This is equivalent to (II-18). The lemma then follows. □

An intuitive explanation of (II-15) is that the slope measured at the multiples of the segments are non-increasing. As will be seen in the later bound derivation process, this propertity greatly simplifies our schedulability bound analysis.

2. R-shaped Workload Constraint Functions

S-shaped workload constraint function is simple and flexible in characterizing workload of different types of tasks, especially those periodic-like ones. In this section, we introduce another type of workload constraint function, namely r-shaped workload constraint function. Formally, we say a function F is r-shaped if $0 < s < t$,

$$F(s)/s \geq F(t)/t, \quad \text{(II-31)}$$

(II-31) means that the rate of the function $F(I)$ is not increasing with I and can be thought as a special case of the s-shaped workload constraint when the step $S \to 0$. Compared with s-shaped functions, r-shaped function is much simpler and this simplicity can greatly reduce the complexity of the schedulability bound analysis.

Three example r-shaped workload constraint functions are given below. Figure 7 is an on-off type workload constraint function which allows a task to release jobs at a constant rate in the "on" mode as long as the task will be remain in "off" mode for a certain time after the "on" mode. Figure 8 is a multi-piece linear workload constraint function which is suitable for tasks that release jobs with higher rates in short time windows and gradually reduces its releasing rate in longer windows. Figure 9 is a workload constraint function in continuous form. Modeling tasks with this type of workload constraint function may have the benefit of analysis simplicity given its smooth property.

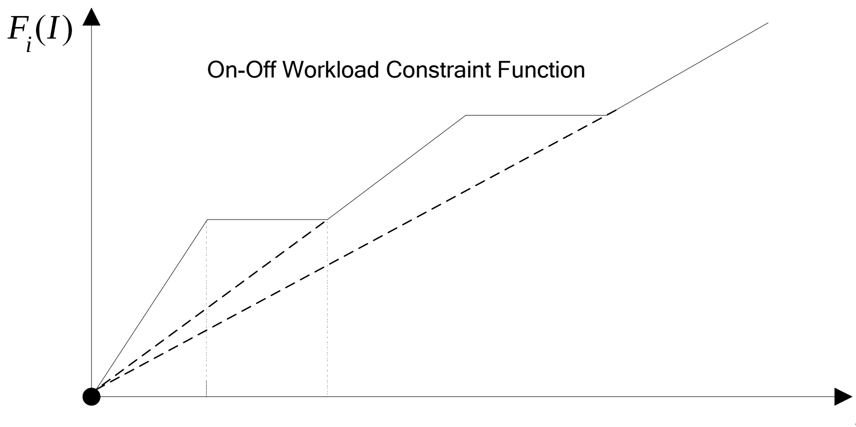

Figure 7. On-off Workload Constraint Function.

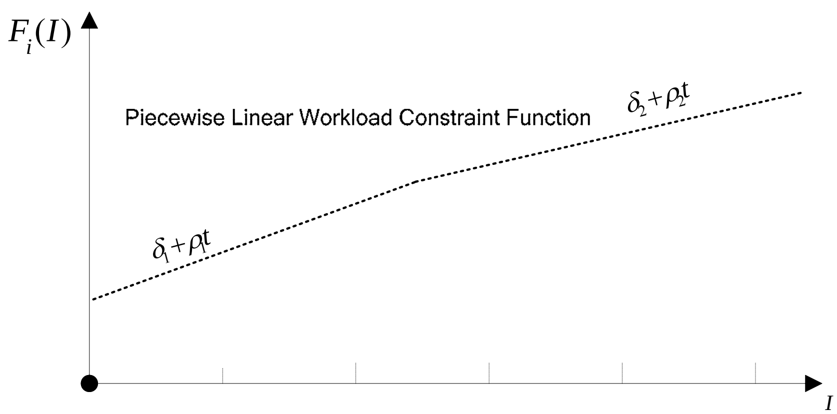

Figure 8. Piecewise Linear Workload Constraint Function.

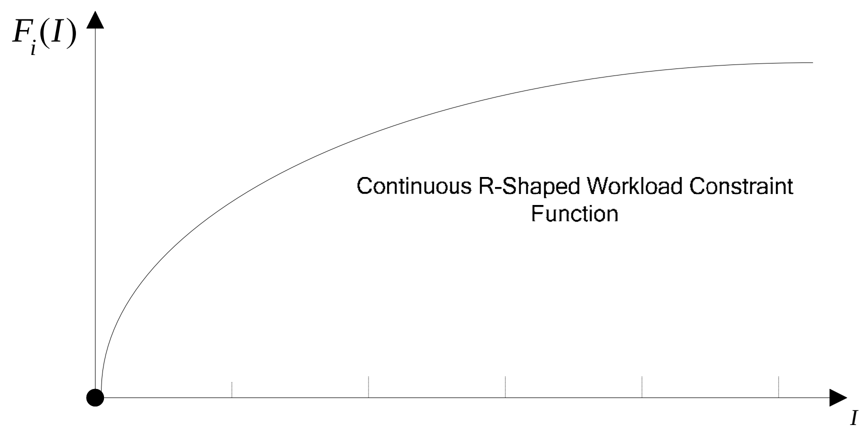

Figure 9. Continuous Workload Constraint Function.

Note that this r-shaped workload constraint function is not a new invention. Similar to the s-shaped workload constraint function, which is a generalization of the classical periodic task model, r-shaped workload constraint function is a generalization of the widely used leaky-bucket traffic model initially proposed in [23] and [24]. R-shaped workload constraint function has been discussed under the name of "star-shaped" function in [16]. But to the best knowledge of the authors, limited work are done on the schedulability bound analysis with this special type of workload constrain function.

C. Service Constraint Functions

With $F(I)$ defined, let us now consider alternatives of $g(t)$. Though one can find many different types of alternatives of $g(t)$, we are only interested in those that can facilitate the delay bound analysis.

Strict Service Constraint Function. We say $G(I)$ is a strict service constraint function for task T if for any $0 \le I \le t$

$$g(t) - g(t-I) \ge G(I)\,. \qquad \text{(II-32)}$$

With this definition, we have $g(I) \ge G(I)$ for all $I \ge 0$ and together with the fact $f(I) \le F(I)$, we have

$$d^* = \sup_{t\ge 0}\left(\inf\left(\tau \mid f(t) \le g(t+\tau)\right)\right) \le \sup_{t\ge 0}\left(\inf\left(\tau \mid F(t) \le G(t+\tau)\right)\right) = \hat{d}\,. \quad \text{(II-33)}$$

That is to say, we can derive an upper bound of d^* based on $F(I)$ and $G(I)$ (which are defined in (II-8) and (II-32), respectively) by using the following expression:

$$d^* \le \sup_{t\ge 0}\left(\inf\left(\tau \mid F(t) \le G(t+\tau)\right)\right). \qquad \text{(II-34)}$$

Note the similarity between the right hand sides of both (II-4) and (II-34). That is, if we substitute $f(t)$ and $g(t)$ in (II-4) by $F(I)$ and $G(I)$ (where $F(I)$ and $G(I)$ are defined in (II-8) and (II-32)), respectively, the right hand side of (II-4) becomes an upper-bound of the worst case delay for task T. As we mentioned earlier, (II-4) has well-understood physical meaning. Thus, if we ought to define any new service constraint function, we would prefer that it satisfies (II-34). Formally, we define Ψ, a preferred class of function G, as follows:

$$\Psi = \{G \mid G \text{ satisfies (3-9) for any given } F(I) \text{ defined in (II-8)}\} \qquad \text{(II-35)}$$

We can define an order over G. For two elements, $G_1 \in \Psi$ and $G_2 \in \Psi$,

$$G_1 \prec G_2, \tag{II-36}$$

if

$$\sup_{t\geq 0}\left(\inf\left(\tau \mid F(t) \leq G_1(t+\tau)\right)\right) \leq \sup_{t\geq 0}\left(\inf\left(\tau \mid F(t) \leq G_2(t+\tau)\right)\right). \tag{II-37}$$

It is obvious that function G defined in (II-32), belongs to Ψ. Also, if we define

$$G^*(t) = 0 \tag{II-38}$$

then,

$$G^*(t) \in \Psi \tag{II-39}$$

That is, Ψ has more than one element. G^* is the maximum element, as for any G in Ψ,

$$G \prec G^*. \tag{II-40}$$

This is because G^* results in a delay upper bound of infinite if we use (II-34) to compute the delay bound.

It would be an interesting and challenging task to carry out a full investigation of G (e.g., its size, its minimum element, etc.). A report on this investigation is yet to be seen. Nevertheless, According to the studies by Parekh-Gallager [64], Chang [18], [19] Cruz [25], and Le Boudec [16], there is at least another element in Ψ. This new element is actually better than (i.e., no larger than) the one defined in (II-32). We introduce this new G as follows.

Generalized Service Constraint Function. $G(I)$ is said to be a generalized service constraint function if for any $t \ge 0$, there exists $I \le t$ that preserves the property

$$g(t) \ge f(t-I) + G(I). \tag{II-41}$$

Typically, we assume that $G(I)$ is non-decreasing and $G(0) \ge 0$. By (II-41) it means that for any t, we can find $t-I$, where $0 \le I \le t$, 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 of jobs have been served, as illustrated in Figure 10.

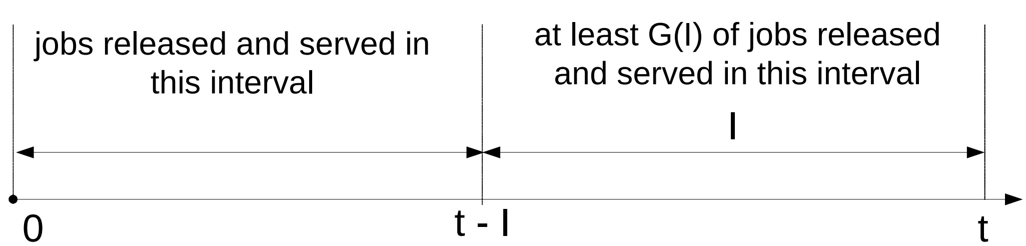

Figure 10. Components in the Generalized Service Constraint Function.

Note that in [16], the authors defined $G(I)$ in form of $g(t) \ge \inf_{0 \le I \le t} \left(f(t-I) + G(I) \right)$. This expression is equivalent to (II-41) except for the case when f and/or G are not continuous. For simplicity, we use (II-41) in this paper. The following theorem proves that the generalized service constraint function defined in

(II-41) is in the preferred class, and is better (tighter delay bound) than the one defined in (3-7).

Theorem 2-1 [16]. Let G' be defined by (II-32) and G'' be defined by (II-35). Given a task with its $F(I)$, we have

$$G'' \in \text{G}. \tag{II-42}$$

and,

$$G'' \prec G'. \tag{II-43}$$

Proof. For any fixed time $t \geq 0$, let $d(t)$ denote the delay of the jobs arrived at time t and let x, $x \geq 0$, be a variable such that

$$x < d(t). \tag{II-44}$$

By definition of delay, we have

$$f(t) > g(t + x). \tag{II-45}$$

By (II-41), we know that for time instant $t + x$, there exists an I0, $0 \leq I_0 \leq t + x$, such that

$$g(t + x) > f(t + x - I_0) + G(I_0). \tag{II-46}$$

By substituting (II-45) into (II-46), we have

$$f(t) > f(t + x - I_0) + G(I_0). \tag{II-47}$$

By (II-41), we know that

$$G(I_0) \geq 0. \tag{II-48}$$

By substituting (II-48) into (II-47), we have

$$f(t) > f(t + x - I_0). \tag{II-49}$$

Since $f(t)$ is non-decreasing, we know that

$$t > t + x - I_0. \tag{II-50}$$

Thus, by (II-8), we have

$$f(t) - f(t + x - I_0) \leq F(I_0 - x). \tag{II-51}$$

Rewrite (II-51) into

$$F(I_0 - x) + f(t + x - I_0) \geq f(x). \tag{II-52}$$

By substituting (II-52) and (II-47), we have

$$F(I_0 - x) \geq G(I_0). \tag{II-53}$$

Rewriting (II-53) with $I_1 = I_0 - x$ as,

$$F(I_1) > G(I_1 + x). \tag{II-54}$$

By plotting the two curves of $F(I)$ and $G(I)$ together, one can notice immediately that the maximum horizontal distance between the two curves is no less than x, or specifically,

$$\sup_{I\geq 0}\left(\inf\left(\tau \mid F(I)\leq G(I+\tau)\right)\right)\geq x. \tag{II-55}$$

Since (II-55) is true for any $x < d(t)$, we know that

$$\sup_{I\geq 0}\left(\inf\left(\tau \mid F(I)\leq G(I+\tau)\right)\right)\geq d(t). \tag{II-56}$$

Since (II-56) is true for any t, $t \geq 0$, we have

$$\sup_{I\geq 0}\left(\inf\left(\tau \mid F(I)\leq G(I+\tau)\right)\right)\geq d^{*}. \tag{II-57}$$

Comparing (II-57) with the definition of preferred class of service constraint function, we know (II-42) is true.

The truefulness of (II-43) is apparent, since we can set the s in the definition generalized service constraint function to be zero. □

From theorem 2-1, we have the following sufficient schedulability test condition.

Corollary 2-1. A task is schedulable if for any $t \geq 0$

$$F(t)\leq G(t+D). \tag{II-58}$$

Proof. By theorem 2-1, if (II-58) holds for all $t \geq 0$, we have $d^{*} \leq D$.

As $G(I)$ defined in (II-41) is the best (the delay bound derived with this G is lowest) by far we have discovered, in the rest of this paper, we will focus on this generalized service constraint function. And hence, we may omit word “generalized” when context is clear.

Now let us consider how to derive the service constraint function for static priority scheduling systems. We say a scheduling system is a static priority one if each of the tasks in the system is assigned a priority value and all the jobs in the task are sharing this value. A job can be executed only if no other jobs with higher priority are waiting for the resource. Without loss of generality, we assume that the tasks are labeled in descending of their priority, i.e. task T_1 has the highest priority and T_n is the task with lowest one. We have the following theorem on the service constraint function of static priority schedulers.

Theorem 2-2. For the static priority scheduling system, $G_i(I)$, a service constraint function for task T_i, is

$$G_i(I) = \sup_{0 \le x \le I} \left(x - \sum_{j=1}^{i-1} F_j(x) \right). \tag{II-59}$$

Proof. We will follow the definition of service constraint function to prove this theorem. Specifically, let t be an arbitrary time instant. We need to prove that there exists a time instant s such that (II-41) holds for the given t as defined in (II-41). We consider the following cases.

Case 1: At time t, T_i is not backlogged. In this case, all released jobs of T_i have been served. Let s equal to t, and we have

$$g_i(t) - f_i(s) = g_i(t) - f_i(t) = 0 = G_i(0)\,. \tag{II-60}$$

Then, by comparing it with (II-41), we have the theorem established.

Case 2: At time t, T_i is backlogged. Let s be the latest time instant before t such that none of the higher priority tasks T_j, $j \le i$, is backlogged. That is, for $j = 1, 2, \ldots, i$,

$$f_j(s) = g_j(s)\,. \tag{II-61}$$

Let t', $s \le t' \le t$, be an arbitrary time instant between s and t inclusively. Since $g_i(t)$ is non-decreasing, we know that the service received by T_i in time interval $[s,\ t]$ should not be less than that in $[s,\ t']$. That is,

$$g_i(t) - g_i(s) \ge g_i(t') - g_i(s). \tag{II-62}$$

By the definition of s, we know that at least one task with priority no lower than T_i is backlogged in $(s,\ t']$, which means that no job from tasks $T_{i+1}, \ldots, T_n$ can be served. So

$$g_i(t') - g_i(s) + \sum\nolimits_{j=1}^{i-1} (g_j(t') - g_j(s)) = t' - s. \tag{II-63}$$

By (II-8) , we have, for $j = 1, 2, \ldots, i$,

$$f_j(t') - f_j(s) \le F_j(t' - s). \tag{II-64}$$

By the definition of s, we know that T_j, j = 1, 2, …, *i*, is not backlogged at time s. Thus, the received service by task T_j, j = 1, 2, …, *i*, in $[s,\ t']$ cannot be more than the total size of the released jobs from T_j in $[s,\ t']$. That is, for *j* = 1, 2, …, *i*,

$$g_j(t') - g_j(s) \le f_j(t') - f_j(s). \tag{II-65}$$

Substituting (II-65) into (II-64), we have, for j = 1, 2, …, *i*,

$$g_j(t') - g_j(s) \le F_j(t' - s). \tag{II-66}$$

Furthermore, substituting (II-66) into (II-63), and we have

$$g_i(t) - g_i(s) + \sum_{j=1}^{i-1} F_j(t' - s) \ge t' - s. \tag{II-67}$$

If we substitute (II-61) into (II-67) and rearrange it, (II-67) will become

$$g_i(t) - f_i(s) \ge t' - s - \sum_{j=1}^{i-1} F_j(t' - s). \tag{II-68}$$

Because t' is an arbitrary time instant in $[s,\ t]$, we have

$$g_i(t) - f_i(s) \ge \max_{0 \le x \le t-s} \left(x - \sum_{j=1}^{i-1} F_j(x) \right). \tag{II-69}$$

The theorem then follows by comparing (II-69) with the definition of service constraint function given in (II-41). □

Note that the sup operation in (II-59) guarantees that $G_i(I)$ is non-negative, and non-decreasing. An intuitive explanation of (II-59) is that the i-th can be blocked by task T_1, T_2, …., T_{i-1} in worst case. Theorem 2-2 can be proved based on the definition of service constraint function. Specifically, given any time instant t, one can define I such that time $t - I$ is the last time instant before t such that a task with priority lower than T_i is scheduled. Then from the property of static priority scheduler, one can prove that task T_i will receive at least $\sup_{0 \le x \le I}\left(x - \sum_{j=1}^{i-1} F_j(x)\right)$ seconds of services in interval [$t - I$, t], and thus the theorem.

D. Workload Rate and Schedulability Bound

1. Workload Rate

Though one can use (II-58) for each task to decide their schedulability test, it maybe time consuming since the (II-58) need to be checked for all $t \ge 0$. In this dissertation, we will take a different approach similar to the utilization bound based test.

Recall that for utilization bound based schedulability test, a task set is schedulable when the utilization of the task set is lower than a pre-derived bound. Our goal here is to develop a similar bound based test algorithm.

Generally speaking, utilization is the resource consumption rate in a measuring time window. For periodic systems, the most effective measuring window is task period. For non-periodic tasks, in [1] and [9], the authors proposed to use the relative deadline of the task as the length of the measuring window in order to define the utilization for non-periodic tasks. While this choice is simple and convenient for some cases, we find that it is too restrictive to meet our goal: a versatile utilization bound analysis system. Instead, we propose to define the length of the measuring window as a linear scale of the relative deadline. That is, the measuring window can be expressed as θD, where $\theta > 0$ is called the scaling parameter and D is the relative deadline of the task.

To avoid confusion with the literature, we refer to this generalized utilization as the scaled workload rate, and it can be formally expressed as

$$W(\theta,\ \Gamma) = \sum_{i=1}^{n} \frac{F_i(\theta D_i)}{\theta D_i}. \tag{II-70}$$

When the context of discussion is clear, the term "scaled" may be omitted. Since $F_i(\theta D_i)$ is an upper bound of the size of jobs that can be released in any time window of length θD_i, $W(\theta,\ \Gamma)$ can be treated as an upper bound of the job releasing rate averaged in a window of length θD_i. Introducing θ into the modeling process parameterizes the utilization measurement. For example, when $\theta = 1$, (II-70) reduces to the definition provided in [1] and [9]. This parameterized measurement of utilization enables flexible representation of different scheduling and workload scenarios, and more importantly, leads to uniform analysis system of schedulability bounds.

2. Schedulability Bound

For a given system, we say $W^*(\theta)$ is schedulability bound if an arbitrary task set Γ is schedulable when the following condition holds:

$$W(\theta,\ \Gamma) < W^*(\theta)\ . \tag{II-71}$$

The challenge is how to derive $W^*(\theta)$ for a broad range of workload patterns and scheduling disciplines. Let the space of all task sets be denoted as Ω, i.e., $\Omega = \{T\}$. Ω can be partitioned into two subsets, Ω_s and Ω_{ns}, where

$$\Omega_s = \{\Gamma \mid \Gamma \text{ is schedulable}\} \tag{II-72}$$

and

$$\Omega_{ns} = \{\Gamma \mid \Gamma \text{ is not schedulable}\}\ . \tag{II-73}$$

$W^*(\theta)$ is a lower bound of the workload rate of these task sets that belong to Ω_{ns}. That is

$$W^*(\theta) = \inf_{\Gamma \in \Omega_{ns}} \left(W(\theta,\ \Gamma)\right) \tag{II-74}$$

Many previous studies have implicitly followed (II-74) to derive schedulability bounds. For example, the minimization process is often achieved by searching along the boundary between Ω_s and Ω_{ns}. Instead of trying to find an analytical representation of the boundary directly, which may be quite challenging task, we will transform all the tasks in Ω, with a special transformation function $Y(\Gamma)$, into a region close to the boundary between Ω_s and Ω_{ns} and then perform a minimization in that region. If the transformation function $Y(\Gamma)$ is properly selected, the bound results obtained on the transformed region may be very close to the bound obtained from the exact boundary between Ω_s and Ω_{ns}, if not the same. Figure 11 illustrates the concept.

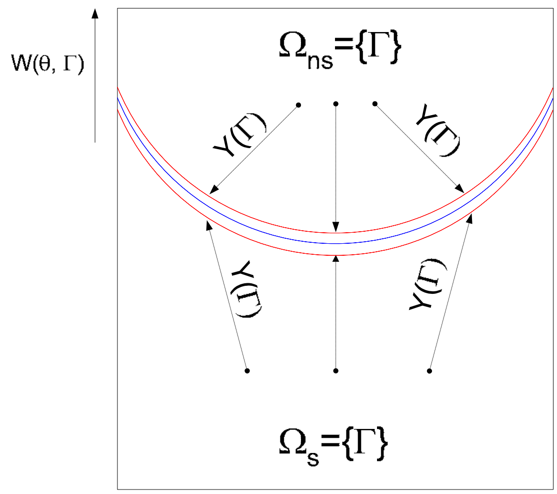

Figure 11. Illustration of the Transformation Function.

To guarantee the correctness of the bound result obtained on the transformed tasks, it is clear that Y needs to satisfies that for any $\Gamma \in \Omega_{ns}$,

$$W(\theta,\ \Gamma) \geq W(\theta,\ \mathrm{Y}(\Gamma)). \tag{II-75}$$

That is, the transformation will not increase the workload rate of any non-schedulable task set. With (II-75) holds, we have

$$W^*(\theta) = \inf_{\Gamma\in\Omega_{ns}}\left(W(\theta,\ \Gamma)\right) \geq \inf_{\Gamma\in\Omega_n}\left(W(\theta,\ \mathrm{Y}(\Gamma))\right). \tag{II-76}$$

Note that (II-75) is only required for non-schedulable tasks. For schedulable tasks, the transformation function may increase its workload rate, but it can be verified that this will not affect the correctness of (II-76).

Now we will show how to construct Y. Let $\alpha_i = \inf_{I>0}\left(G_i(I+D)/F_i(I)\right)$ and h be the value of j at which $(\alpha_i - 1)\dfrac{F_j(\theta D_j)}{\theta D_j}$ is minimized. That is, for $i = 1,\ 2,\ ...,\ n$,

$$(\alpha_h - 1)\frac{F_h(\theta D_h)}{\theta D_h} \le (\alpha_i - 1)\frac{F_i(\theta D_i)}{\theta D_i}, \qquad \text{(II-77)}$$

Let $\{T_1',\ T_2',\ ...,\ T_n'\}$ denote the transformed task set $Y(\Gamma)$, F_i' the workload constraint function, and D_i' the relative deadline for T_i', $i = 1,\ 2,\ ...,\ n$. We construct the function Y as follows, for $i = 1,\ 2,\ ...,\ n$,

$$D_i' = D_i, \qquad \text{(II-78)}$$

for all $i \ne h$

$$F_i'(I) = F_i(I), \qquad \text{(II-79)}$$

and for $i = h$,

$$F_i'(I) = \alpha_h F_i(I). \qquad \text{(II-80)}$$

That is, the function Y changes the F_h to $\alpha_h F_i(I)$ and keeps the other F_i and D_i untouched. Now, we will find a lower bound of the schedulability bound with the help of the just constructed transformation function. By (II-70), we have

$$W(\theta,\ Y(\Gamma)) = \sum_{j=1}^{n} \frac{F_j(\theta D_j)}{\theta D_j} + (\alpha_h - 1)\frac{F_h(\theta D_h)}{\theta D_h}, \tag{II-81}$$

By Lemma 2-1, for any $\Gamma \in \Omega_{ns}$, there must exist a task T_j such that $\alpha_j < 1$. By substituting $\alpha_j < 1$ into (II-81), we get, for any $\Gamma \in \Omega_{ns}$,

$$(\alpha_h - 1)\frac{F_h(\theta D_h)}{\theta D_h} \le 0\,. \tag{II-82}$$

By substituting (II-82) into (II-81), we have, for any $\Gamma \in \Omega_{ns}$,

$$W(\theta,\ \Upsilon(\Gamma)) \le \sum_{j=1}^{n} \frac{F_j(\theta D_j)}{\theta D_j} = W(\theta,\ \Gamma)\,. \tag{II-83}$$

Furthermore, by substituting (II-83) into (II-74), we get

$$W^*(\theta) \ge \inf_{\Gamma\in\Omega_{ns}} \left(W(\theta,\ \Upsilon(\Gamma))\right) \ge \inf_{\Gamma\in\Omega} \left(W(\theta,\ \Upsilon(\Gamma))\right). \tag{II-84}$$

Then, by substituting (II-81) into (II-84), we have the following theorem about the workload bound.

Theorem 2-3. A lower bound of schedulability bound for a scheduler with respect to Ω is given by

$$W^*(\theta) = \inf_{\Gamma\in\Omega}\left(\sum_{j=1}^{n}\frac{F_j(\theta D_j)}{\theta D_j} + \min_{i=1,\,2,\,\ldots,\,n}\left((\alpha_i - 1)\frac{F_i(\theta D_i)}{\theta D_i}\right)\right), \qquad \text{(II-85)}$$

where $\alpha_i = \inf_{I>0}\left(G_i(I+D)/F_i(I)\right)$.

Proof. By the definition of schedulability bound, we just need to prove that $\forall\Gamma\in\Omega$, Γ is schedulable if

$$W(\theta,\,\Gamma) < W^*(\theta), \qquad \text{(II-86)}$$

where $W^*(\theta)$ is given in (II-91). We will prove the theorem by contradiction. Let Γ be a task set such that (II-86) holds, yet Γ is not schedulable. Let T_i denote a non-schedulable task in Γ. Then from Corollary 2-1, there exists s such that

$$F_i(s) > G_i(s+D_i). \qquad \text{(II-87)}$$

Thus,

$$\alpha_i < 1. \qquad \text{(II-88)}$$

Since $F_i(\theta D_i) \ge 0$, we have

$$\frac{F_i(\theta D_i)}{\theta D_i} \ge \alpha_i \frac{F_i(\theta D_i)}{\theta D_i}. \qquad \text{(II-89)}$$

By adding $\sum_{j=1,\,j\ne i}^{n}\frac{F_j(\theta D_j)}{\theta D_j}$ on both sides of (II-89), we have

$$\sum_{j=1}^{n}\frac{F_j(\theta D_j)}{\theta D_j} \geq \sum_{j=1,\, j\neq i}^{n}\frac{F_j(\theta D_j)}{\theta D_j} + \alpha_i \frac{F_i(\theta D_i)}{\theta D_i}\,. \qquad \text{(II-90)}$$

This contradicts with (II-86). Then follows the theorem. □

Note that Theorem 2-3 is a general result. It is a closed-form representative of the schedulability bound with task workload and service constraint functions being input parameters. It holds for any workload constraint function and any work conserving scheduler. By substituting specific forms of $F_i(I)$ and $G_i(I)$ into (II-85) and solving the optimization problem, one can obtain the schedulability bounds for different schedulers.

For scheduler with s-shaped workload constraint functions, we have the following result on its schedulability bound.

Corollary 2-2. Given any static priority scheduler with s-shaped tasks, a schedulability bound with respect to Ω is given by

$$W^*(\theta) = \min_{\Gamma\in\Omega}\left(\min_{i=1,\,2,\,\ldots,\,n}\left(\sum_{j=1,\, j\neq i}^{n}\frac{F_j(\theta D_j)}{\theta D_j} + \min_{m=0,\,1,\,\ldots}\left(\frac{G_i(m\cdot S_i + D_i)}{\theta D_i + m\cdot \min\left\{\theta D_i, S_i\right\}}\right)\right)\right). \qquad \text{(II-91)}$$

Corollary 2-2 is a specialization of Theorem 2-3 for s-shaped workload constraint functions and can be proved by plug-in the properties of s-shaped workload constraint function into (II-85).

Proof. By close observation of (II-91) and (II-85), we only need to prove

$$\min_{I>0}\left(\frac{G_i(I+D)}{F_i(I)}\right)\frac{F_i(\theta D_i)}{\theta D_i} \geq \min_{m=0,\,1,\,\ldots}\left(\frac{G_i(mS_i+D_i)}{\theta D_i + m\cdot\min\left\{\theta D_i, S_i\right\}}\right). \qquad \text{(II-92)}$$

Rewrite (II-92) it as

$$\min_{I>0}\left(\frac{G_i(I+D)}{F_i(I)}\frac{F_i(\theta D_i)}{\theta D_i}\right) \geq \min_{m=0,\,1,\,\ldots}\left(\frac{G_i(mS_i+D_i)}{\theta D_i + m\cdot\min\left\{\theta D_i, S_i\right\}}\right). \qquad \text{(II-93)}$$

We will prove (II-93) in two cases: $\theta D_i \leq S_i$ and $\theta D_i > S_i$.

Case 1: $\theta D_i \leq S_i$**.** For any $I > 0$, let $\ell = \lfloor I / S_i \rfloor$. Then, by (II-11), $F_i(I) \leq F_i((\ell+1)S_i)$, and thus,

$$\frac{G_i(I+D)}{F_i(I)} \geq \frac{G_i(I+D)}{F_i((\ell+1)S_i)}. \qquad \text{(II-94)}$$

By Lemma 2-1, we have $F_i((\ell+1)S_i) \leq (\ell+1)F_i(S_i)$. Thus, (II-94) becomes

$$\frac{G_i(I+D)}{F_i(I)} \geq \frac{G_i(I+D)}{(\ell+1)F_i(S_i)}. \qquad \text{(II-95)}$$

Since $\theta D_i \leq S_i$, by (II-11), we have $F_i(S_i) = F_i(\theta D_i)$, and hence

$$\frac{G_i(I+D)}{F_i(I)} \geq \frac{G_i(I+D)}{(\ell+1)F_i(\theta S_i)}. \qquad \text{(II-96)}$$

Rewrite (II-96) as

$$\frac{G_i(I+D)}{F_i(I)}\frac{F_i(\theta S_i)}{\theta S_i} \geq \frac{G_i(I+D)}{(\ell+1)\theta S_i}. \qquad \text{(II-97)}$$

Since $G_i(I+D)$ is non-decreasing, we have $G_i(I+D) \geq G_i(\ell S_i + D)$. Hence (II-99) becomes

$$\frac{G_i(I+D)}{F_i(I)}\frac{F_i(\theta S_i)}{\theta S_i} \geq \frac{G_i(\ell S_i + D)}{(\ell+1)\theta S_i}. \qquad \text{(II-98)}$$

Minimizing the right hand of (II-97) will not invalidate (II-97), we have

$$\frac{G_i(I+D)}{F_i(I)}\frac{F_i(\theta S_i)}{\theta S_i} \geq \min_{m=0,\ 1,\ \ldots}\left(\frac{G_i(mS_i + D)}{(m+1)\theta S_i}\right). \qquad \text{(II-99)}$$

Since (II-98) holds for any $I > 0$, we have

$$\min_{I>0}\left(\frac{G_i(I+D)}{F_i(I)}\frac{F_i(\theta S_i)}{\theta S_i}\right) \geq \min_{m=0,\ 1,\ \ldots}\left(\frac{G_i(mS_i + D)}{(m+1)\theta S_i}\right). \qquad \text{(II-100)}$$

Thus the corollary is established for this case.

Case 2: $\theta D_i > S_i$. Let $h = \lceil \theta D_i / S_i \rceil$. It is clear that $h \geq 1$ and,

$$\frac{F_i(\theta D_i)}{\theta D_i} = \frac{F_i(hS_i)}{\theta D_i}. \qquad \text{(II-101)}$$

For any $I > 0$, let $\ell = \lfloor I / S_i \rfloor$. From Lemma 2-1, we know that $F_i(hS_i) \geq \ell F_i((h+\ell)S_i)/(h+\ell)$. Hence, we can rewrite (II-101) as

$$\frac{F_i(\theta D_i)}{\theta D_i} \geq \frac{hF_i((h+\ell)S_i)}{(h+\ell)\theta D_i}. \tag{II-102}$$

Since F_i is non-decreasing and $h \geq 1$, we have

$$F_i((h+\ell)S_i) \geq F_i((\ell+1)S_i). \tag{II-103}$$

Substituting (II-103) into (II-102), we have

$$\frac{F_i(\theta D_i)}{\theta D_i} \geq \frac{hF_i((\ell+1)S_i)}{(h+\ell)\theta D_i}. \tag{II-104}$$

By (II-11) and the definition of I, we have

$$F_i((\ell+1)S_i) \geq F_i(I) \tag{II-105}$$

Substituting (II-105) into (II-104), we get

$$\frac{F_i(\theta D_i)}{\theta D_i} \geq \frac{hF_i(I)}{(h+\ell)\theta D_i}. \tag{II-106}$$

By multiplying $\dfrac{G_i(I+D)}{F_i(I)}$ on both sides of (II-106), we have

$$\frac{G_i(I+D)}{F_i(I)} \frac{F_i(\theta D_i)}{\theta D_i} \geq \frac{hG_i(I+D)}{(h+\ell)\theta D_i}. \tag{II-107}$$

Since $hS_i \geq \theta D_i$, we have

$$\frac{hS_i}{\theta D_i} \geq \frac{hS_i + \ell S_i}{hP_i + \theta D_i}. \tag{II-108}$$

By substituting (II-108) into (II-107), we get

$$\frac{G_i(I+D)}{F_i(I)} \frac{F_i(\theta D_i)}{\theta D_i} \geq \frac{G_i(I+D)}{hS_i + \theta D_i}. \tag{II-109}$$

Since $G_i(I+D)$ is non-decreasing, we have $G_i(I+D) \geq G_i(\ell S_i + D)$. Hence (II-99) becomes

$$\frac{G_i(I+D)}{F_i(I)} \frac{F_i(\theta S_i)}{\theta S_i} \geq \frac{G_i(\ell S_i + D)}{hP_i + \theta D_i}. \tag{II-110}$$

Minimizing the right hand side of (II-109) will not invalidate (II-109). So

$$\frac{G_i(I+D)}{F_i(I)} \frac{F_i(\theta D_i)}{\theta D_i} \geq \min_{m=0,\,1,\,\ldots} \left(\frac{G_i(mS_i + D)}{mS_i + \theta D_i} \right). \tag{II-111}$$

Since (II-111) holds for any $I > 0$, we have

$$\min_{I>0} \left(\frac{G_i(I+D)}{F_i(I)} \frac{F_i(\theta D_i)}{\theta D_i} \right) \geq \min_{m=0,\,1,\,\ldots} \left(\frac{G_i(mS_i + D)}{mS_i + \theta D_i} \right). \tag{II-112}$$

Then follows the corollary. □

Similarly, for a system with set of tasks with r-shaped workload constraint functions, we have the following result on its schedulability bound.

Corollary 2-3. A lower bound of the utilization bounds with r-shaped arrival curves is

$$U^*(\theta) = \inf_{\Gamma\in\Omega}\left(min_{i=1\ldots n}\left\{ min_{I\geq 0}\left\{ \sum_{j=1\ldots n}^{j\neq i}\frac{F_j(\theta D_j)}{\theta D_j} + \frac{G_i(I+D_i)}{I+\theta D_i}\right\}\right\}\right). \tag{II-113}$$

Proof. If a task $T_i \in \Gamma$ is not schedulable, then by Corollary 2-1, there must exists a time instant $s \geq 0$ such that $F_i(s) > G_i(s+D_i)$. Combine with the fact that F_i is r-shaped, we have:

$$\frac{F_i(\theta D_i)}{\theta D_i} \geq \frac{F_i(s+\theta D_i)}{s+\theta D_i} \geq \frac{G_i(s+D_i)}{s+\theta D_i}. \tag{II-114}$$

By adding $\sum_{j=1\ldots n}^{j\neq i}\frac{F_j(\theta D_j)}{\theta D_j}$ to both sides, we get

$$\sum_{j=1\ldots n}^{j\neq i}\frac{F_j(\theta D_j)}{\theta D_j} + \frac{F_i(\theta D_i)}{\theta D_i} \geq \sum_{j=1\ldots n}^{j\neq i}\frac{F_j(\theta D_j)}{\theta D_j} + \frac{G_i(s+D_i)}{s+\theta D_i}. \tag{II-115}$$

Since the left hand side of the above inequality is $U(\theta, \Gamma)$ and the right hand is always greater then $U^*(\theta)$, we know $U(\theta, \Gamma) \geq U^*(\theta)$. In summary, if a task set Γ is not schedulable, then its workload is always greater then $U^*(\theta)$, and we prove the corollary.

CHAPTER III

SCHEDULABILITY BOUND FOR STATIC PRIORITY SCHEDULERS

A. Static Priority Schedulers

We make the following assumptions on the static priority scheduling system under consideration: (1) each task is assigned a static priority, and (2) the scheduler performs preemptive, priority based task scheduling. To simplify the notation, we further assume that tasks are labeled in descending priority order, i.e. T_1 has the highest priority and T_n has the lowest one. Obviously, the static priority scheduler is work conserving.

In our model, we do not assume that priorities are assigned in any particular order, and deadline inversion is allowed. To measure the impact of deadline inversion to individual tasks, we formally define λ_1, the degree of deadline inversion for task T_i, as follows:

$$\lambda_i = \frac{\max_{j=1 \ldots i} \left\{ D_j \right\}}{D_i}. \tag{III-1}$$

Note that $\lambda_i \geq 1$. When $\lambda_i = 1$, deadlines of tasks with priority higher than T_i are less than or equal to D_i. Hence, no deadline inversion occurs to T_i. When $\lambda_i > 1$, deadline inversion occurs to task T_i. Taking one step further, we let λ be the degree of deadline inversion for the whole task set task set:

$$\lambda = \max \left\{ \lambda_1, \lambda_2, \ldots, \lambda_n \right\} \tag{III-2}$$

For example, let us consider a three-task task set $\Gamma = \{T_1,\ T_2,\ T_3\}$ with D_1 =10, D_2 =5, and D_3 =20. T_1 has the highest priority and T_3 has the lowest one. Now by (III-1), we can calculate the deadline inversion for T_1, T_2, and T_3 as follows:

$$\lambda_1 = D_1 / D_1 = 1\,, \tag{III-3}$$

and

$$\lambda_2 = \max\left(D_1,\ D_2\right) / D_2 = \max\left(10,\ 5\right) / 5 = 2\,, \tag{III-4}$$

and

$$\lambda_3 = \max\left(D_1,\ D_2,\ D_3\right) / D_3 = \max\left(10,\ 5,\ 20\right) / 20 = 1\,. \tag{III-5}$$

By (III-2), we have

$$\lambda = \max\left(\lambda_1,\ \lambda_2,\ \lambda_3\right) = 2\,. \tag{III-6}$$

That is to say, this system is having a deadline inversion of 2.

The introduction of λ into the modeling process parameterizes the static priority scheduling algorithms. For example, when $\lambda = 1$, the scheduler becomes deadline monotonic. Furthermore, for periodic tasks, when $\lambda = 1$, and $D_i = P_i$, the scheduler becomes rate monotonic. In this case, if we set $\theta = 1$ in computing the workload rate, by (II-11) and (II-70) we get

$$W(1,\ \Gamma) = \sum_{i=1}^{n} \frac{F_i(P_i)}{P_i} = \sum_{i=1}^{n} \frac{C_i}{P_i}. \tag{III-7}$$

The left side of (III-7) is the classical utilization definition for periodic task set. This example demonstrates the generality and flexibility of our task and scheduler models. We will see that these properties play an important role in our schedulability analysis technique.

B. Schedulability Bound For S-Shaped Tasks

In previous section, we have introduced the deadline inversion parameter that can capture the difference between various priority assignment schemes. In this section, we will first introduce two additional parameters, normalized deadline and task set heterogeneity, which are related to s-shaped tasks and then derive a parameterized schedulability based on these three key parameters.

1. Schedulability Bound

For task T with a given s-shaped constraint function, we define its normalized deadline, k, as follows:

$$k = D/S, \tag{III-8}$$

where D is the relative deadline of T and S is the segment parameter in the constraint function. k can be viewed as the deadline using S as the measurement unit, and it characterizes tightness of the deadline requirements. The smaller the k, the more difficult it is to schedule the task. It will be clear later that parameter k plays a critical role in schedulability test. Following the convention in the literature [5], [34], [43], [49], [50], [52], [56], [57], [66], [75], and [80], we assume that all the tasks have the same value of k. That is, for $i = 1,\ 2,\ ...,\ n$

$$k_i = k\ . \qquad \text{(III-9)}$$

Recall that the workload rate $W(\theta,\ \Gamma)$ is an upper bound of the job releasing rate averaged in a window of finite length θD_i. Since an s-shaped function allows bust job releasing in short widows, as long as the job releasing rate slows down in longer window lengths, using the workload rate measured in a window of length θD_i for schedulability bound analysis may over-estimate the actual resource demand of the task, and thus result in under-estimated bounds. To overcome this problem, in our schedulability bound analysis, we want to take account into consideration the variance of the workload rate measured in different windows. We proposed a task parameter, workload heterogeneity $\eta_i(\hbar)$ for this purpose, which is defined as,

$$\eta_i(\hbar) = \frac{F_i(\hbar S_i)/\hbar}{F_i((\hbar+1)S_i) - F_i(\hbar S_i)}, \qquad \text{(III-10)}$$

where $\hbar$ is a positive integer. Intuitively, $V_i(\hbar)$ is the ratio between the workload rate measured in [0, $\hbar S_i$], and the one measured in [$\hbar S_i, (\hbar+1)S_i$] as shown in Figure 12. Clearly, for periodic task, $\eta_i(\hbar) =$ 1.

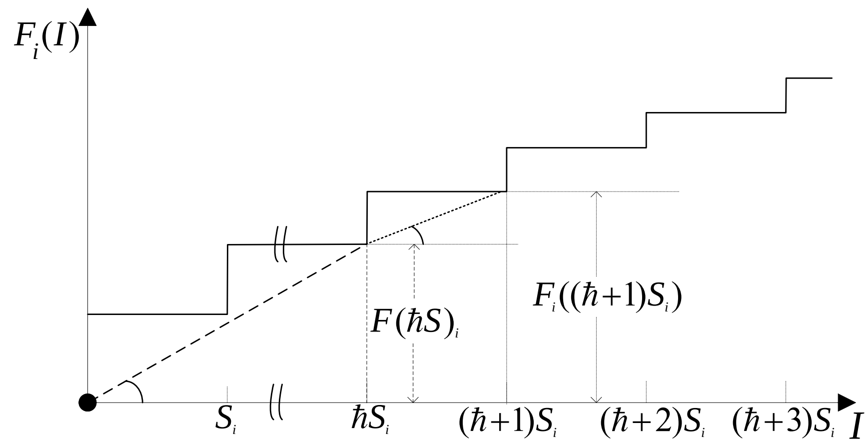

Figure 12. Workload Heterogeneity.

Taking one step further, we define the heterogeneity of a task set as

$$\eta(\hbar,\ \Gamma) = \min\left(\eta_1(\hbar),\ \eta_2(\hbar),\ \ldots,\ \eta_n(\hbar)\right). \qquad \text{(III-11)}$$

Note that the heterogeneity is first introduced in [56] and [57] in the form of C^1 / C^2 for a special case where $\hbar = 1$.

By substituting (II-59) into (II-91), and then optimize the consequent inequality for the different parameters, we have the following closed-form schedulability bound for static priority scheduler.

Theorem 3-4. Given a Static Priority Scheduler and collection of tasks with s-shaped workload constraint functions, a schedulability bound with $\theta = 1/\lambda$ is given by

$$W^*\left(\frac{1}{\lambda}\right) = \begin{cases} 1 & h \le \frac{r}{r+1} \\ \frac{r}{h}\left(n\left(\left(\frac{r+1}{r}h\right)^{\frac{1}{n}} - 1\right) + 1 - h\right) & \frac{r}{r+1} < h \le 1 \\ rn\left(\left(\frac{r+1}{r}\right)^{\frac{1}{n}} - 1\right) & h = 2, 3, \ldots \end{cases} \qquad \text{(III-12)}$$

where k is the normalized deadline defined in (III-8), $r = k\eta$, θ is the heterogeneity of tasks defined in (III-11), and h is

$$h = \frac{k}{\lambda}. \qquad \text{(III-13)}$$

Note that with our selection of $\theta = 1/\lambda$, the workload is measured in a window length of $\theta D = D/\lambda = kS/\lambda = hS$. As suggested when we deal with (III-11), we will measure the heterogeneity in the same window as that used for workload rate. That is, following (III-11), we have

$$\eta_i(h) = \begin{cases} C_i^2 / C_i^1 & h < 1 \\ \dfrac{\left(\sum_{j=1}^{h-1} C_i^j\right) \Big/ h-1}{C^h} & h = 2,\ 3,\ .... \end{cases} \qquad \text{(III-14)}$$

and $\eta = \min(\eta_1,\ \eta_2,\ ...,\ \eta_n)$.

Proof. See Appendix B.

Theorem 3-4 defines a multi-dimensional schedulability bound surface based on four system parameters, i.e. deadline inversion ratio, λ, normalized deadline k, task set heterogeneity η, and number of tasks n.

2. Evaluation of Bound

On the basis of the theoretical results developed in previous section, we evaluate the system performance in this section, using the schedulability bound as the primary performance measure. Note that chance of a newly task being admitted into the system is proportional to the schedulability bound, and thus high schedulability bounds are preferred. The following factors affect the schedulability bound:

- The normalized deadline, k , of a task (see (III-8) for its definition). When normalized deadlines of tasks become tighter, the expected schedulability bound will be lower.
- The heterogeneity η of a task (see (III-11) for its definition). This parameter gauges fitness of the workload constraint function in capturing the diversity of job sizes in a task. One may use task heterogeneity to improve resource allocation, rather than the pessimistic assumption of only using the worst case job size in the periodic model. The system performance is expected to improve with the increase of η value.
- The degree of deadline inversion, λ (see (III-2), for its definition). λ indicates the degree of deadline inversion in a priority assignment. The system performance suffers when the λ value increases.

The sensitivity of the schedulability bound with respect to the three key factors, k , η , and λ is analyzed, and the results are plotted in Figure 13. The 3-D graphs of schedulability bounds vs. the three parameters are examined for 1,000 tasks, i.e., n = 1,000. In each of the three figures, for a fixed λ value, we varied k from 10^{-4} to 10^{4} and η from 1 to 102 to make the following observations:

- As expected, the tighter the deadlines, the lower the schedulability bounds. The sensitivity is especially significant when k is small (i.e., less than 5). For example, in Figure 13.(a), when k changes from 0.5 to 5, the schedulability bound increases from 0.50 to 0.912. The sensitivity becomes less significant when k is large. In Figure 13.(a), when k changes from 5 to 100, the schedulability bound increases only about 10% (0.995 - 0.912).

- Large heterogeneity of task leads to improved schedulability bounds. The sensitivity is higher when η is small, (i.e., 10), and becomes less sensitive for larger η values. For example, for $\lambda = 1$, k = 1, when η changes from 1 to 10, the schedulability bound increases from 0.693 to 0.953. When η changes from 10 to 100, the schedulability bound increases from 0.953 to 0.995.
- As the degree of deadline inversion increases, the schedulability bound decreases. Consider a point where k = 1 and η = 1 in all the three figures. When λ changes from 1, 4, to 16 (Figures 13.(a)-(c)), the schedulability bound decreases from 0.693, to 0.250 and then to 0.063.

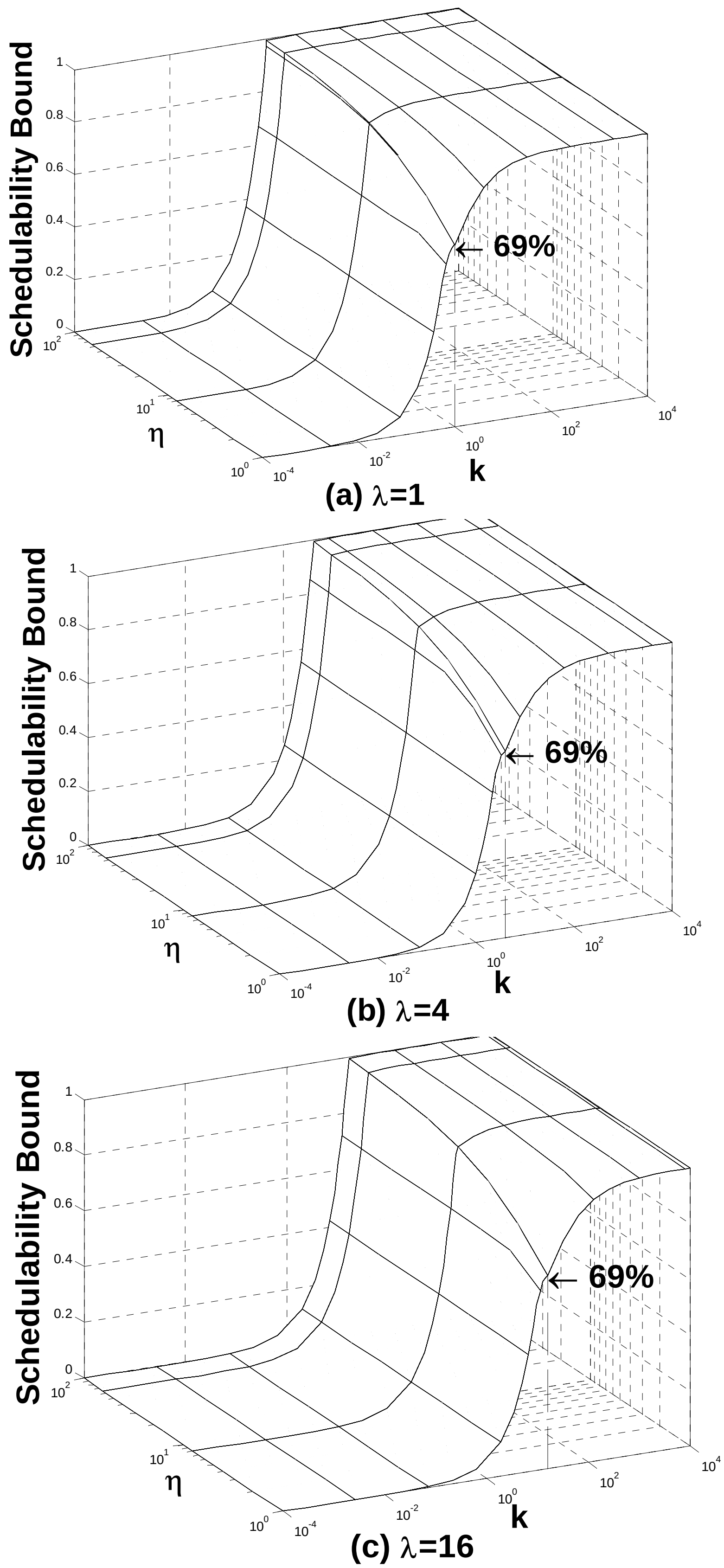

Figure 13. Schedulability Bound of Static Priority Schedulers.

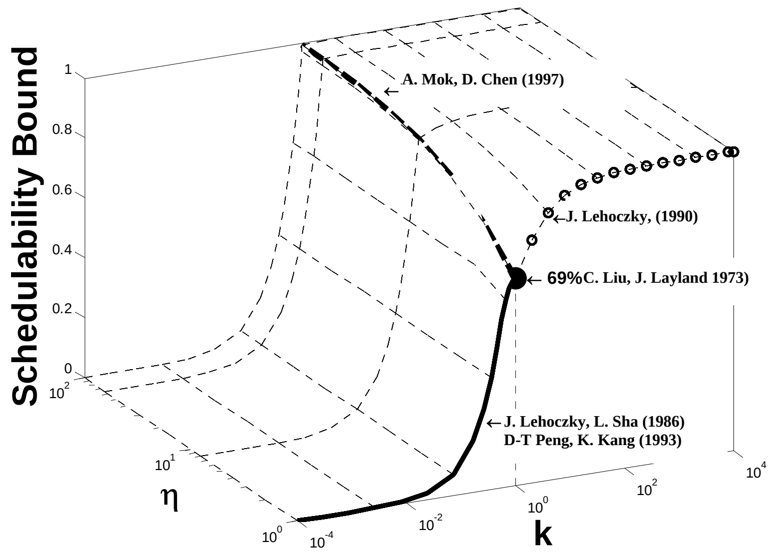

Figure 14. Comparison of New Schedulability Bounds with Previous Results.

In the above discussions, we examined how the three factors, k , η , and λ independently affect the system performance. By further observing Theorem 3-4, one notice that W^* depends on k/λ . This implies a tradeoff relationship between k and λ . To improve the schedulability bound, one can either increase the normalized deadline, or lower the priority inversion ratio. On the other hand, we can keep the schedulability unchanged while adjusting both k and λ as long as we can let k/λ be constant. To our knowledge, this is the first time that the tradeoff relationship between deadline and deadline inversion is explicitly expressed in an analytical form.

Theorem 3-4 is highly flexible. It gives the schedulability bounds for a wide range of systems by parameterization of normalized deadline, heterogeneity of task, and degree of deadline inversions. In this subsection, we illustrate how to match our results with those derived in previous studies.

- First, we consider the classical periodic system, in which the task's deadlines are equal to task periods, and tasks are scheduled by a rate monotonic scheduler. In [49], C. Liu, and J. Layland derived a schedulability bound as follows:

$$U^* = n\left(2^{\frac{1}{n}} - 1\right) . \qquad \text{(III-15)}$$

- Now, we re-derive (III-15) by using Theorem 3-4. Note that by (III-7), the workload rate reduces to the classical definition of utilization in this case. This system has no deadline inversion, and hence λ = 1. Furthermore, since this is a periodic system, from (III-7) we have η = 1. Because the relative deadline is equal to the length of period, k = 1. Substituting λ = 1, η = 1, and k = 1 into (III-12), we have

$$W^*(1) = U^* = n\left(2^{\frac{1}{n}} - 1\right) . \qquad \text{(III-16)}$$

- As n approaches infinity, U^* and W^* will approach 69%. This becomes one point in the graph of Figure 14 we redraw the graph in Figure 13.(a) to highlight the match.
- Second, we analyze the periodic system in which task's deadlines are less than periods and tasks are scheduled by a rate monotonic scheduler. In [43] and [66], J.

P. Lechoczky, L. Sha, D.-T. Peng, and K. G. Shin derived an utilization bound as follows:

$$U^* = \begin{cases} 1/k & k \le \frac{1}{2} \\ n\left((2k)^{\frac{1}{n}} - 1\right) + 1 - k & \frac{1}{2} < k \le 1 \end{cases} . \qquad \text{(III-17)}$$

- By (III-7), in this case the workload rate and the classical utilization has the following relationship

$$W^*(1) = \frac{U^*}{k} . \qquad \text{(III-18)}$$

We now re-derive (III-18) with Theorem 3-4. This system has no deadline inversion, and hence λ = 1. From (III-7) we have η = 1. Because the relative deadlines are less than the period lengths, we know that k < 1. Substituting λ = 1, and η = 1 into (III-12), we get

$$W^*(1) = \begin{cases} 1 & k \le \frac{1}{2} \\ \frac{1}{k}\left(n\left((2k)^{\frac{1}{n}} - 1\right) + 1 - k\right) & \frac{1}{2} < k \le 1 \end{cases} . \qquad \text{(III-19)}$$

By applying (III-18) into (III-19), and we have

$$U^* = \begin{cases} k & k \le \frac{1}{2} \\ n\left((2k)^{\frac{1}{n}} - 1\right) + 1 - k & \frac{1}{2} < k \le 1 \end{cases} . \qquad \text{(III-20)}$$

This is exactly the same as (III-19). We illustrate this result by a curve in the 3-D graph of in Figure 14.

- Third, we analyze the periodic system in which the task's deadlines are multiples of periods and tasks are scheduled by a rate monotonic scheduler. In [29], J. P. Lechoczky obtained an utilization bound as follows:

$$U^* = k(n-1)\left((\tfrac{k+1}{k})^{\frac{1}{n-1}} - 1\right). \qquad \text{(III-21)}$$

Note that by (III-7), the workload rate reduces to the classical utilization in this case. We will re-derive (III-21) through Theorem 3-4. This system has no deadline inversion, and hence λ = 1. Furthermore, as this is a periodic system, we have η = 1. Because the relative deadline is composed of multiple periods, we have k being an integer larger than 1. Substituting λ = 1 and η = 1 into (III-12), we have

$$W^*(1) = kn\left((\tfrac{k+1}{k})^{\frac{1}{n}} - 1\right). \qquad \text{(III-22)}$$

Note that (III-22) is not as tight as (III-21) (n vs. $n-1$). This is due to the fact that Theorem 3-4 is obtained for general s-shaped functions. However, the exact bound of (III-21) can be easily obtained with our general method, as discussed Appendix D. In Figure 14, a curve illustrates this match of results.

- Fourth, we analyze the AM multi-frame system in which task's deadlines are equals to periods and tasks are scheduled by a rate monotonic scheduler. In [56] and [57], A. Mok and D. Chen obtained an utilization bound as follows:

$$U^* = an\left((\tfrac{a+1}{a})^{\frac{1}{n}} - 1 \right) , \qquad \text{(III-23)}$$

where $a = \min_{i=1,\ 2,\ldots,\ n}\{E_i^0 / E_i^1\}$, and the E_i^j is the job size of the j-th job. By definition, we know that $a = r$, and by (III-7), the workload rate is the same as the utilization used in [56] and [57]. Now we re-derive (III-23) from Theorem 3-4. In our terminology, this system has no deadline inversion, and hence λ = 1. Furthermore, as this is a multi-frame system, from (III-7) we have $\eta = r$. Because the relative deadline equals to periods, we have k = 1. Substituting λ = 1, $\eta = r$, and k = 1 into (III-12), we get

$$W^*(1) = rn\left((\tfrac{r+1}{r})^{\frac{1}{n}} - 1 \right) , \qquad \text{(III-24)}$$

which is exactly the same as (III-23). Again, this match is drawn on Figure 14.

- Finally, we examine a system that uses the general real-time model defined in [56] and [57]. The task's deadlines are equal to periods and tasks which are scheduled by a rate monotonic scheduler. In [56] and [57], A. Mok and D. Chen obtained an utilization bound as follows:

$$U^* = rn\left((\tfrac{r+1}{r})^{\frac{1}{n}} - 1 \right) . \qquad \text{(III-25)}$$

Again, by (III-7), we know that the workload rate reduces to the classical utilization. Recall that s-shaped workload constraint functions reduce to the general real-time task model when $L = \infty$. Then it is trivial to show (III-25) is a special case of Theorem 3-4.

Through the simple algebraic analysis mentioned above, we have illustrated that the results obtained in [34], [43], [49], [56], and [57] are special cases of Theorem 3-4. We virtually match all the previous results, with the exception of the slight difference between (III-21) and (III-22). Furthermore, our results cover many cases that have not been analyzed before, because the results from [34], [43], [49], [56], and [57] are merely one point and three curves in the 3-D graph shown in Figure 14.

3. Extensions

In [22], the authors proved that the utilization bound for periodic task improves when the periods of tasks are divisible. This observation is applicable to Theorem 3-4 as stated next.

Corollary 3-4. Given static priority system and a task set Γ with s-shaped workload constraint functions Γ, Γ is schedulable if

$$W^*\left(\frac{1}{\lambda}\right)=\begin{cases}\min\left(1,\ \frac{\lambda}{k}\eta\left(n'\left((\frac{\eta+1}{\eta}\frac{k}{\lambda})^{\frac{1}{n'}}-1\right)+1-\frac{k}{\lambda}\right)\right) & k\le\lambda;\\ \frac{k}{\lambda}\eta n'\left((\frac{k\eta+\lambda}{k\eta})^{\frac{1}{n'}}-1\right) & k=\Delta\lambda,\ \text{and } \Delta \text{ is a positive integer.}\end{cases}$$

(III-26)

where n' is the number tasks with non-dividable segment lengths.

Proof. See Appendix E.

In a non-preemptive system, a high priority task T_i can be blocked by a lower priority task T_j, $j > i$, for a length of $J_{\max}$, the maximum job size of T_j. Thus, this non-preemption effect will lead to a priority inversion for the length of $J_{\max}$. During this interval, the system acts as if T_j has a higher priority than T_i. In the worst case, the system in this interval is operating in a mode with $\lambda = \lambda^* = \max_{i=1,\,2,\,...,\,n}\{D_i\} / \min_{i=1,\,2,\,...,\,n}\{D_i\}$. By substituting this into Theorem 3-4, we obtain the schedulability bound for non-preemption case.

Corollary 3-5. Given a non-preemptive static priority scheduler and a task set Γ with s-shaped workload constraint functions, Γ is schedulable if

$$W^*\left(\frac{1}{\lambda^*}\right) = \begin{cases} \min\left(1,\ \dfrac{\lambda^*}{k}\eta\left(n'\left((\dfrac{\eta+1}{\eta}\dfrac{k}{\lambda^*})^{\frac{1}{n'}} - 1\right) + 1 - \dfrac{k}{\lambda^*}\right)\right) & k \le \lambda^*; \\ \dfrac{k}{\lambda^*}\eta n'\left((\dfrac{k\eta+\lambda^*}{k\eta})^{\frac{1}{n'}} - 1\right) & k = \Delta\lambda^*, \text{ and } \Delta \text{ is a positive integer.} \end{cases} \qquad \text{(III-27)}$$

where $\lambda^* = \max_{i=1,\,2,\,...,\,n}\{D_i\} / \min_{i=1,\,2,\,...,\,n}\{D_i\}$ and n' is the number tasks with non-dividable segment lengths.

Proof. It is apparent based on the above analysis.

C. Schedulability Bound for R-Shaped Tasks

1. Schedulability Bound

Theorem 3-5. Given a static priority scheduler and tasks with r-shaped workload constraint functions, a schedulability bound is $W^*(1) = 1/\lambda$.

Proof. By Theorem 2-2, we know $G_i(t)$ is a service constraint function to T_i, where

$$G_i(I) = \max_{0 \le x \le I}\left(x - \sum_{j=1}^{i-1} F_j(x)\right). \quad \text{(III-28)}$$

By (III-28) and Corollary 2-2, we have,

$$W^*(1) \ge \min_{i=1,\,2,\,\ldots,\,n}\left(\sum_{j=1}^{i-1}\frac{F_j(D_j)}{D_j} + \min_{I>0}\left(\frac{\min_{0 \le x \le I+D_i}\left(x - \sum_{j=1}^{i-1} F_j(x)\right)}{I + D_i}\right)\right). \quad \text{(III-29)}$$

Since reducing the range of the max operation in $max_{0 \le x \le I+D_i}\left\{x - \sum_{j=1}^{i-1} F_j(x)\right\}$ will not increase its value, we have

$$\max_{0 \le x \le I+D_i}\left(x - \sum_{j=1}^{i-1} F_j(x)\right) \ge \max\left(0,\ I + D_i - \sum_{j=1}^{i-1} F_j(I + D_i)\right). \quad \text{(III-30)}$$

By substituting (III-30) into (III-29), we have,

$$W^*(1) \ge \min_{i=1,\,2,\,\ldots,\,n}\left(\min_{I>0}\left(\sum_{j=1}^{i-1}\frac{F_j(D_j)}{D_j} + \frac{\max\left(0,\ I + D_i - \sum_{j=1}^{i-1} F_j(I + D_i)\right)}{I + D_i}\right)\right) \quad \text{(III-31)}$$

By definition of r-shaped workload constraint function (II-31), we have

$$\frac{F_j(D_i)}{D_i} \geq \frac{F_j(I+D_i)}{I+D_i}\,. \tag{III-32}$$

By substituting (III-32) into (III-31), we have

$$W^*(1) \geq \min_{i=1,\,2,\,\ldots,\,n}\left(\sum_{j=1}^{i-1}\frac{F_j(D_j)}{D_j} + \max\left(0,\ 1-\sum_{j=1}^{i-1}\frac{F_j(D_i)}{D_i}\right)\right). \tag{III-33}$$

From the definition of deadline inversion (III-1) and (III-2), we have

$$D_i \leq \lambda D_j\,. \tag{III-34}$$

By substituting (III-34) into (III-33), we get

$$W^*(1) \geq \min_{i=1,\,2,\,\ldots,\,n}\left(\sum_{j=1}^{i-1}\frac{F_j(D_j)}{D_j} + \max\left(0,\ 1-\sum_{j=1}^{i-1}\frac{\lambda F_j(D_j)}{D_j}\right)\right). \tag{III-35}$$

Re-arrange (III-35) into

$$W^*(1) \geq \min_{i=1,\,2,\,\ldots,\,n}\left(\max\left(\sum_{j=1}^{i-1}\frac{F_j(D_j)}{D_j},\ 1-(\lambda-1)\sum_{j=1}^{i-1}\frac{F_j(D_j)}{D_j}\right)\right). \tag{III-36}$$

It can be verified that

$$\max\left(\sum_{j=1}^{i-1}\frac{F_j(D_j)}{D_j},\ 1-(\lambda-1)\sum_{j=1}^{i-1}\frac{F_j(D_j)}{D_j}\right) \geq \frac{1}{\lambda}\,. \tag{III-37}$$

Then by substituting (III-37) into (III-36), we have,

$$W^*(1) \geq \min_{i=1,\, 2,\, \ldots,\, n} (1/\lambda) = 1/\lambda . \qquad \text{(III-38)}$$

Then the theorem is proven □

2. Evaluation of the Bound

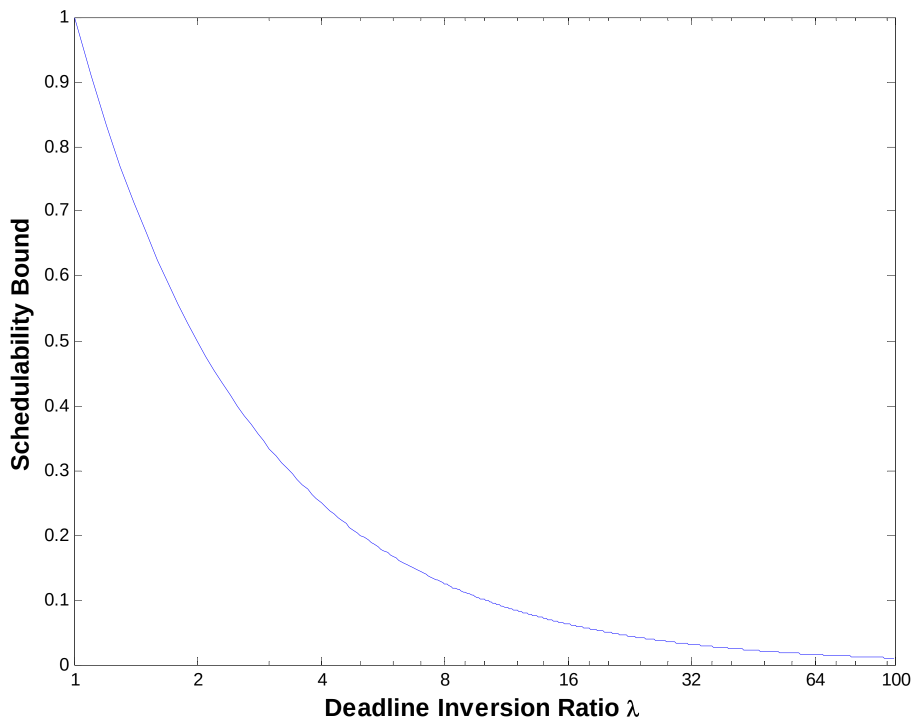

Figure 15. Schedulability Bound of Static Priority Schedulers with R-Shaped Tasks.

Figure 15 plots the schedulability bound of static priority scheduler against the deadline inversion ratio. From the figure, one can noticed that the schedulability bound monotonically decreases with the increasing of deadline inversion ratio. In other word, deadline monotonic scheduler is optimal among all static priority schedulers in terms of schedulability bounds, i.e. deadline monotonic scheduler has the highest bound value, 100.0%, because its deadline inversion is 1. In practical systems, to achieve high system resource utilization, one can minimize the deadline inversions, e.g. assign higher priority to tasks with lower relative deadline values.

By a close observation of Theorem 3-5, one can also notice that the schedulability bound does not depend on the number of tasks. This feature makes this bound result very scalable for large-scale dynamic systems since when the number of tasks changes, the schedulability test algorithm does not need to re-calculate the schedulability bound.

One may wonder why the schedulability bound in Theorem 3-5 does not have a "normalized deadline" parameter, like the "k" used in Theorem 3-4 that measures the tightness of deadline requirement. Does this mean the deadline assignment play no role in the schedulability test? The answer is no. Note that in Theorem 3-5, the scaling parameter θ, is 1. That is to say, the length of the measuring window of the workload rate is D_i for task T_i. When the deadline requirements are relaxed, i.e. D_i increases, the length of the measuring window of the workload will also increase. Recall the definition of r-shaped workload constraint function. We know that its workload rate is non-increasing with the increase of the length of measuring window. Hence, the workload rate measured for the tasks will possibly decrease with the relaxation of deadline requirements. As a result of this, the total measured workload rate of a task set may reduce. The effect is the same as the increasing of the schedulability bound.

3. Extensions

We can extend Theorem 3-5 to non-preemptive systems using the same strategy as the one used in Corollary 3-5.

Corollary 3-6. Given a non-preemptive static priority scheduler and a task set Γ with r-shaped workload constraint functions, Γ is schedulable if $W\left(1,\ \Gamma\right) = 1/\lambda^*$ where $\lambda^* = \max_{i=1,\ 2,\ \ldots,\ n}\{D_i\}/\min_{i=1,\ 2,\ \ldots,\ n}\{D_i\}$.

Proof. In a non-preemptive system, a high priority task T_i can be blocked by a lower priority task T_j, $j > i$, for a length of $J_{\max}$, the maximum job size of T_j. Thus, this non-preemption effect will lead to a priority inversion for the length of $J_{\max}$. During this interval, the system acts as if T_j has a higher priority than T_i. In the worst case, the system in this interval is operating in a mode with $\lambda = \lambda^* = \max_{i=1,\ 2,\ \ldots,\ n}\{D_i\}/\min_{i=1,\ 2,\ \ldots,\ n}\{D_i\}$. By substituting this into Theorem 3-5, we have the corollary. □

A different approach to handle non-preemptive system is to calculate the workload rate using a measuring window length of D_i - $J_{\max}$, or specifically,

$$W(1,\ \Gamma) = \sum_{i=1}^{n} \frac{F_i(D_i - J_{\max})}{D_i - J_{\max}}. \tag{III-39}$$

Since in a non-preemptive system, compared with preemptive one, a high priority job can be blocked by a lower priority one by at most $J_{\max}$ time unit, reduce the relative deadline requirements of each task by the $J_{\max}$ will guarantee that any schedulable task in a preemptive system will also be schedulable in a non-preemptive one.

CHAPTER IV

SCHEDULABILITY BOUND FOR WEIGHTED ROUND ROBIN SCHEDULERS

A. Weighted Round Robin Schedulers

In this chapter, we analyze the schedulability bounds for the weighted round robin schedulers which arrange tasks into a circle and serve them in round robin fashion. A token is typically passed among the tasks and a task (job) can execute up to H_i time units once receives the token, where H_i is called the allocation/bandwidth of the tasks. If the task does not have any job waiting for the processor resource, or if it has already ran for H_i time units, the token will be passed to the next task in the circle. Typically H_i is calculated as

$$H_i = O_i \cdot \left(TTRT - \tau\right) \ , \tag{IV-1}$$

where O_i, $0 \le O_i \le 1$, is the weight of task T_i, and the $TTRT$, the target token rotation time which is the desired time to finish one round of token rotation, and τ, the protocol overhead, which is the time speet on token transversal and other protocol and network operations. Typically,

$$\sum_{i=1}^{n} O_i = 1 \ . \tag{IV-2}$$

Generally speaking, large $TTRT$ means longer waiting time since a task have to wait longer before receiving the token and starting execution. On the other hand, small $TTRT$

may lead to higher system overhead, e.g. context switching cost. To capture the effect of $TTRT$ on schedulability bound, we introduce a parameter, normalized token rotation frequency γ. Formally,

$$\gamma = \left\lfloor \frac{D_{\min}}{TTRT} \right\rfloor , \tag{IV-3}$$

where $D_{\min} = \min(D_i)$ is the shortest relative deadline of all the tasks in the system. γ can be considered as the measured token rotation frequency in a window of length $D_{\min}$. Higher frequency typically leads to shorter scheduling delay and thus higher schedulability bound.

For convenience, we define another α, the protocol overhead ratio, as follows:

$$\alpha = \frac{\tau}{TTRT} . \tag{IV-4}$$

Theorem 4-6. For an arbitrary weighted round robin scheduler, a service constraint function for task T_i, is

$$G_i(I) = \left\lfloor \frac{I}{\sum_{j=1}^{n} H_j} \right\rfloor H_i . \tag{IV-5}$$

Proof. We prove this theorem based on the definition of service constraint function. Let t be an arbitrary time instant. If at time t, all the jobs from task T_i have been served, then we can let $s = t$ and (II-41) is true. Now we focus the case that at time t, task T_i is backlogged. Let s be the last time before t such that T_i is not backlogged. That is to say, at time s, we have

$$g_i(s) = f_i(s). \tag{IV-6}$$

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

$$g_i(t) - g_i(s) \geq \left\lfloor \frac{I}{\sum_{j=1}^{n} H_j} \right\rfloor H_i. \tag{IV-7}$$

By substituting (IV-6) into (IV-7), we have

$$g_i(t) - f_i(s) \geq \left\lfloor \frac{I}{\sum_{j=1}^{n} H_j} \right\rfloor H_i. \tag{IV-8}$$

By comparing (IV-8) with (II-41), we know that the theorem is true. □

B. Schedulability Bound for Normalized Weighted Assignment

In a normalized assignment scheme, the allocation H_i is assigned as follows:

$$H_i = \frac{W_i(1)}{W(1,\ \Gamma)} \cdot \left(TTRT - \tau \right) , \tag{IV-9}$$

where

$$W_i(1) = \frac{F_i(D_i)}{D_i} \ , \qquad \text{(IV-10)}$$

and

$$W(1,\ \Gamma) = \sum_{j=1}^{n} W_j(1) \ , \qquad \text{(IV-11)}$$

and $TTRT$ is the target token rotation time and is a system constant. $TTRT$ can be treated as the desired time needed to finish one round of token rotation.

Theorem 4-7. For an arbitrary weighted round robin scheduler with a collection of task set Γ, a lower bound of schedulability bound is:

$$W^*(1) = \min_{T\in\Omega}\left(\min_{i=1,\ 2,\ \ldots,\ n}\left(\min_{I\geq 0}\left(\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor \frac{W_i(1)\cdot\left(TTRT-\tau\right)}{F_i(I)}\right)\right)\right). \qquad \text{(IV-12)}$$

Proof. By (II-58), we know that for any T_i in Γ, is schedulable as long as, for any $I \geq 0$

,

$$\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor \frac{W_i(1)\cdot\left(TTRT-\tau\right)}{W(1,\ \Gamma)} \geq F_i(I) \, . \qquad \text{(IV-13)}$$

Rewrite (IV-13) as

$$W(1,\ \Gamma) \leq \left\lfloor \frac{I+D_i}{TTRT}\right\rfloor \frac{TTRT-\tau}{F_i(I)} W_i(1) \, . \qquad \text{(IV-14)}$$

It is easy to see that

$$\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor \frac{TTRT-\tau}{F_i(I)} W_i(1) \le \min_{T\in\Omega}\left(\min_{i=1,\ 2,\ \ldots,\ n}\left(\min_{I\ge 0}\left(\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor \frac{W_i(1)\cdot\left(TTRT-\tau\right)}{F_i(I)} \right)\right)\right) \ . \tag{IV-15}$$

By substituting (IV-15) into (IV-14), we know that T_i is schedulable as long as

$$W(1,\ \Gamma) \le \min_{T\in\Omega}\left(\min_{i=1,\ 2,\ \ldots,\ n}\left(\min_{I\ge 0}\left(\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor \frac{W_i(1)\cdot\left(TTRT-\tau\right)}{F_i(I)} \right)\right)\right). \tag{IV-16}$$

Then we established the theorem. □

1. Schedulability Bound for S-Shaped Tasks

In this subsection, we will derive a schedulability bound for the weighted robin scheduler with normalized weight assignment scheme based on three system parameters: normalized deadline defined in (III-8), normalized token rotation frequency defined in (IV-3), protocol overhead ratio defined in (IV-4) and workload burstness μ .We define the workload burstness for a task T_i as

$$\mu_i = \frac{F(S_i)/S_i}{F(\lceil k \rceil S_i)/(\lceil k \rceil S_i)} \ , \tag{IV-17}$$

That is to say, the burstness is the ratio between the workload rate measured in a window size of S_i and the one measured in $\lceil k \rceil S_i$. The greater the ratio, the more bursty the workload is. We define the *task set workload burstness* as

$$\mu = \max_{i=1,\ 2,\ \ldots,\ n}(\mu_i) \ . \tag{IV-18}$$

By (II-15), we know that

$$\mu \geq 1 \ . \tag{IV-19}$$

With the parameters defined, we have the following schedulability bound result.

Theorem 4-8. A lower bound of schedulability bound for weighted round robin scheduler with normalized weight assignment, and s-shaped tasks is given by

$$W\left(\lceil k \rceil / k,\ \Gamma\right) = \frac{1}{1/\gamma + 1} \cdot \left(1 - \alpha\right) \cdot \frac{1}{\mu} \cdot \min\left(1,\ k\right). \tag{IV-20}$$

Proof. By (IV-12), we know that a schedulability bound is

$$W(1,\ \Gamma) = \min_{T \in \Omega}\left(\min_{i=1,\ 2,\ \ldots,\ n}\left(\min_{I \geq 0}\left(Z(i)\right)\right)\right). \tag{IV-21}$$

where

$$Z(i) = \max_{I \geq 0}\left(\left\lfloor \frac{I + D_i}{TTRT} \right\rfloor \frac{TTRT - \tau}{F_i(I)} W_i(1)\right). \tag{IV-22}$$

Rewrite (IV-22) into

$$Z(i) = (1-\alpha)\cdot \max_{I\geq 0}\left(\frac{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor}{\frac{I+D_i}{TTRT}} \frac{I+D_i}{F_i(I)} \frac{F_i(D_i)}{D_i} \right). \quad \text{(IV-23)}$$

Since $\frac{I+D_i}{TTRT} \leq \left\lfloor \frac{I+D_i}{TTRT} \right\rfloor + 1$, we have

$$Z(i) = (1-\alpha)\cdot \max_{I\geq 0}\left(\frac{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor}{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor + 1} \frac{I+D_i}{F_i(I)} \frac{F_i(D_i)}{D_i} \right). \quad \text{(IV-24)}$$

It is easy to verify that

$$\frac{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor}{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor + 1} = \frac{1}{1+\frac{1}{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor}} \geq \frac{1}{1+\frac{1}{\left\lfloor \frac{D_{\min}}{TTRT} \right\rfloor}} = \frac{1}{1/\gamma + 1}. \quad \text{(IV-25)}$$

By substituting (IV-25) into (IV-24), we have

$$Z(i) \geq (1-\alpha)\cdot \max_{I\geq 0}\left(\frac{1}{1/\gamma+1} \frac{I+D_i}{F_i(I)} \frac{F_i(D_i)}{D_i} \right). \quad \text{(IV-26)}$$

Now let $I = mS_i + \omega$ where $0 \leq \omega < S_i$. By (II-11), we have

$$F_i(I) \leq F_i((m+1)S_i). \quad \text{(IV-27)}$$

By substituting (IV-27) into (IV-26) and rearrange it, we have

$$Z(i) \geq \left(1-\alpha\right) \cdot \frac{1}{1/\gamma+1} \max_{I \geq 0} \left(\frac{mS_i + \omega + D_i}{F_i\left((m+1)S_i\right)} \frac{F_i(D_i)}{D_i} \right). \tag{IV-28}$$

Since $\omega \geq 0$, we have

$$Z(i) \geq \left(1-\alpha\right) \cdot \frac{1}{1/\gamma+1} \max_{I \geq 0} \left(\frac{mS_i + D_i}{F_i\left((m+1)S_i\right)} \frac{F_i(D_i)}{D_i} \right). \tag{IV-29}$$

By (II-15) we have

$$\frac{F_i\left((m+1)S_i\right)}{(m+1)S_i} \leq \frac{F_i\left(S_i\right)}{S_i}. \tag{IV-30}$$

By substituting (IV-30) into (IV-29) and rearrange it, we get

$$Z(i) \geq \left(1-\alpha\right) \cdot \frac{1}{1/\gamma+1} \max_{I \geq 0} \left(\frac{mS_i + D_i}{(m+1)F_i\left(S_i\right)} \frac{F_i(D_i)}{D_i} \right). \tag{IV-31}$$

By definition of s-shaped workload constraint function (II-11), we know that

$$F_i(D_i) = F_i(kS_i) = F_i(\lceil k \rceil S_i). \tag{IV-32}$$

By substituting (IV-32) into (IV-31), we have

$$Z(i) \geq \left(1-\alpha\right) \cdot \frac{1}{1/\gamma+1} \max_{I \geq 0} \left(\frac{mS_i + D_i}{(m+1)F_i\left(S_i\right)} \frac{F_i(\lceil k \rceil S_i)}{kS_i} \right). \tag{IV-33}$$

Rewrite (IV-33) into

$$Z(i) \geq (1-\alpha) \cdot \frac{1}{k} \frac{1}{1/\gamma+1} \frac{F_i(\lceil k \rceil S_i)}{F_i(S_i)} \max_{m=0,\ 1,\ 2,\ \ldots} \left(\frac{m+k}{m+1} \right). \tag{IV-34}$$

It can be verified that

$$\max_{m=0,\ 1,\ 2,\ \ldots} \left(\frac{m+k}{m+1} \right) \geq \min(1,\ k). \tag{IV-35}$$

By substituting (IV-35) into (IV-34), we have

$$Z(i) \geq (1-\alpha) \cdot \frac{1}{k} \frac{1}{1/\gamma+1} \frac{F_i(\lceil k \rceil S_i)}{F_i(S_i)} \min(1,\ k). \tag{IV-36}$$

By (IV-17)and (IV-18), we have

$$\frac{F_i(\lceil k \rceil S_i)}{F_i(S_i)} = \frac{\lceil k \rceil}{\mu_i} \geq \frac{\lceil k \rceil}{\mu}\ . \tag{IV-37}$$

By substituting (IV-37) into (IV-36), we get

$$Z(i) \geq (1-\alpha) \cdot \frac{1}{1/\gamma+1} \frac{\lceil k \rceil}{k\mu} \min(1,\ k). \tag{IV-38}$$

By substituting (IV-38) into (IV-21), we have a schedulability bound of

$$W^*(1,\ \Gamma) = (1-\alpha) \cdot \frac{1}{1/\gamma+1} \frac{\lceil k \rceil}{k\mu} \min(1,\ k). \tag{IV-39}$$

By (II-11), we know that

$$W(1,\ \Gamma)=\sum_{i=1}^{n}\frac{F_i(D_i)}{D_i}=\frac{\lceil k\rceil}{k}\sum_{i=1}^{n}\frac{F_i(\lceil k\rceil S_i)}{\lceil k\rceil S_i}=\frac{\lceil k\rceil}{k}\sum_{i=1}^{n}W\left(\frac{\lceil k\rceil}{k},\ \Gamma\right). \quad \text{(IV-40)}$$

By substituting (IV-40) into (IV-39), we have a schedulability bound with scaling parameter $\theta=\lceil k\rceil/k$ in form of

$$W^*(\lceil k\rceil/k,\ \Gamma)=(1-\alpha)\cdot\frac{1}{1/\gamma+1}\cdot\frac{1}{\mu}\cdot\min(1,\ k). \quad \text{(IV-41)}$$

Then we have the theorem proven. □

By a close observation of (IV-20), one can notice:

a) Schedulability bound monotonically decreases with the increasing of protocol overhead ratio.

b) Given protocol overhead ratio α, normalized deadline k and workload burstness μ, the schedulability bound increases with the increasing of normalized token rotation frequency and attains its maximum $(1-\alpha)\min(1,\ k)/\mu$ when $\gamma\to\infty$ which implies infinitely fast token rotation and is corresponding to the well-known theoretical generalized processor sharing scheduler (GPS). Figure 16 illustrates this trend for the case $k=1$.

c) Given protocol overhead ratio α, normalized deadline k and normalized token rotation frequency γ, the schedulability bound decreases with the increasing of task set workload burstness μ. This is because increasing of μ implies more bursty workloads which are typically more difficult to schedule than less bursty ones. The schedulability bound is maximized when $\mu = 1$ which corresponds to periodic tasks. Figure 17 illustrates this trend for the case of $\gamma = 2$ which means token rotates twice per D_{min} time interval.

d) Given protocol overhead ratio α, task set workload burstness μ and normalized token rotation frequency γ, relaxing deadline requirements (increasing k) improves schedulability bound when $k \leq 1$, but has no effect when $k > 1$. This trend is shown in Figure 18. We should note that this does not imply that, when $k > 1$, relaxing deadline requirement has no effect on schedulability test, because, according to (IV-10) and (II-12), the calculated workload rate in the window D_i may decrease with the increasing of k. This decreasing of workload rate is equivalent to the increasing of schedulability bound.

e) For periodic tasks with relative deadlines equal to their period (D_i=P_i), and token rotation frequency $\gamma = 1$, (token rotates at least once per D_{min} interval), protocol overhead ratio $\alpha = 0$, the schedulability bound is 50.0%. This is highlighted on Figure 16 as a point.

f) For periodic tasks with relative deadlines equal to their period (D_i=P_i), and normalized token rotation frequency $\gamma = 2$, (token rotates at least twice per D_{min} interval), protocol overhead ratio $\alpha = 0$, the schedulability bound is 66.7%. When the normalized deadline increases from 1 to 1.5, the schedulability bound stays at 66.7%. This is illustrated as a serial of points in

Figure 17. Note this does not mean that *k* has no effect on schedulability test since the workload rate is measured differently as explained in (c).

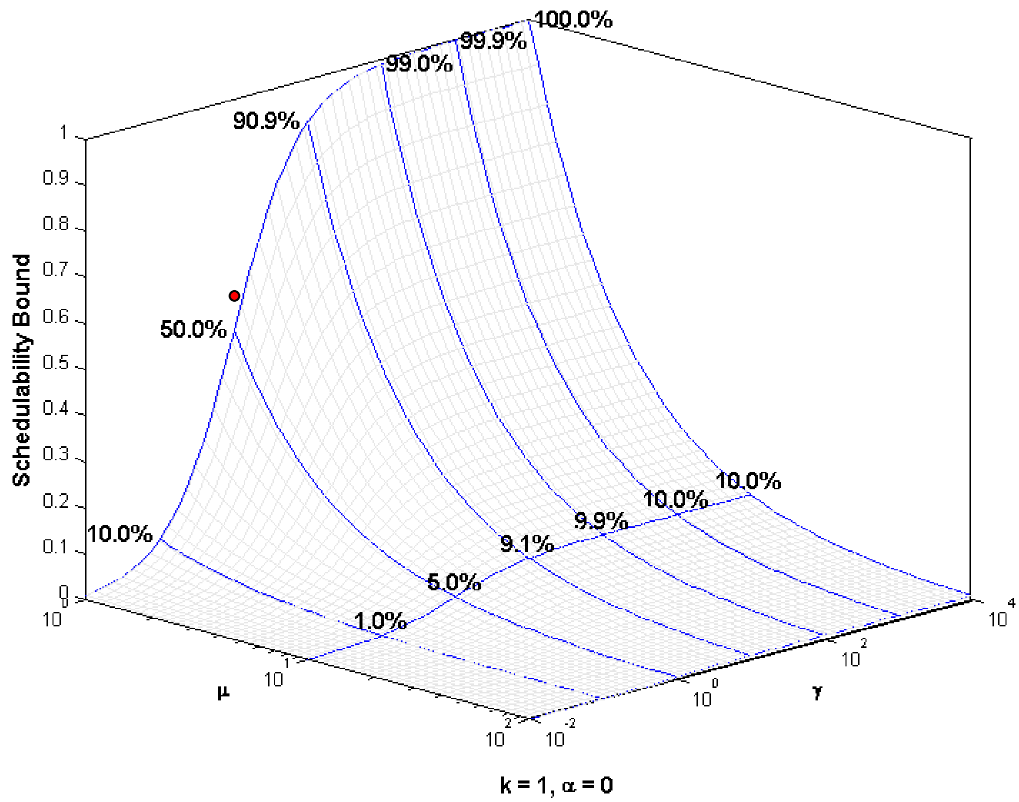

Figure 16. Schedulability Bound of Weighted Robin Scheduler with Normalized Weight Assignment and Fixed Normalized Deadline $k = 1$.

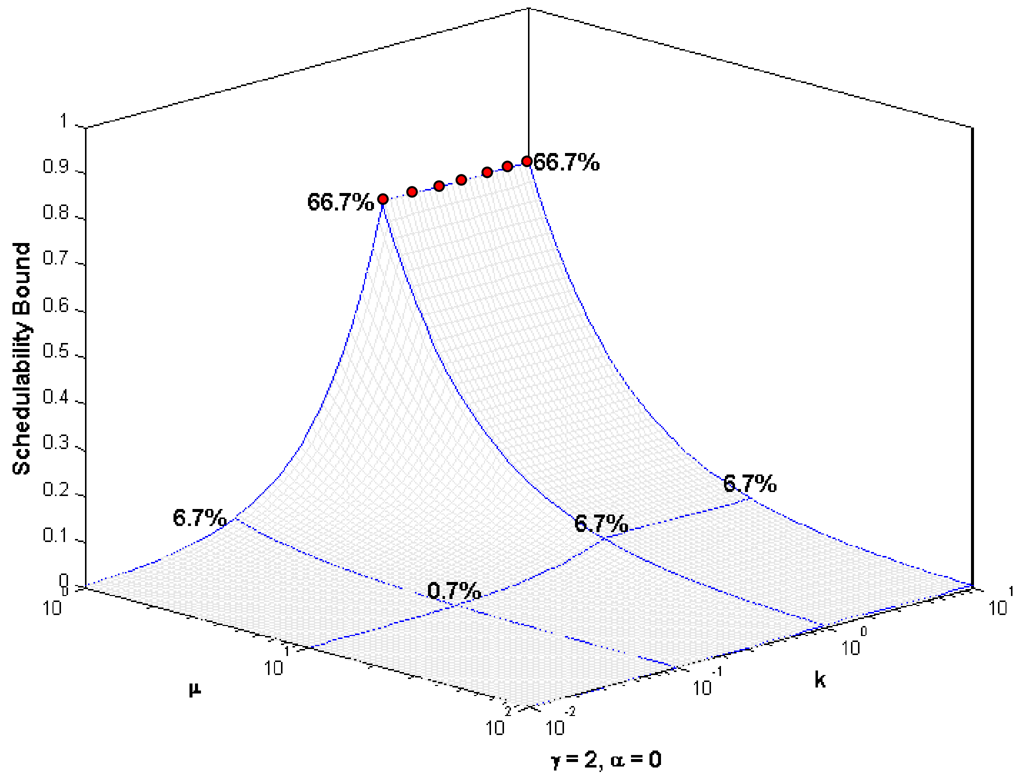

Figure 17. Schedulability Bound of Weighted Robin Scheduler with Normalized Weight Assignment and Fixed Normalized Token Rotation Frequency $\gamma = 2$.

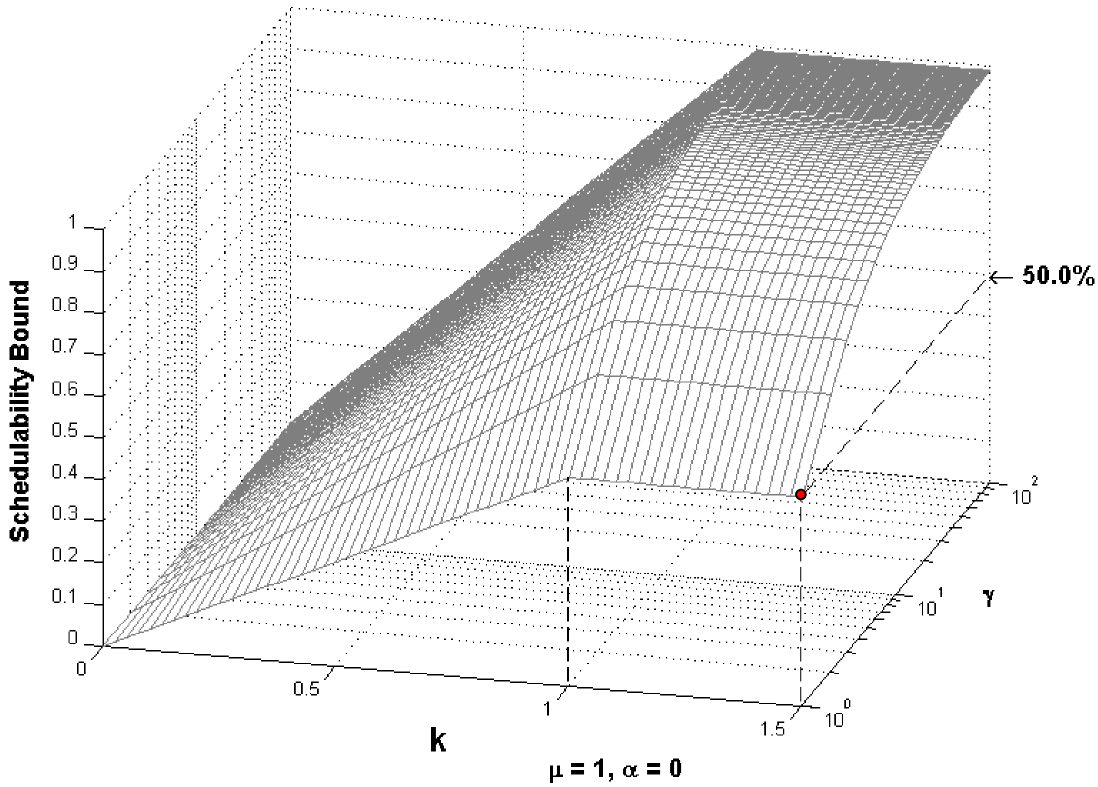

Figure 18. Schedulability Bound of Weighted Robin Scheduler with Normalized Weight Assignment and Fixed Task Set Workload Burstness $\mu = 1$.

Corollary 4-7. A lower bound of schedulability bound for weighted round robin scheduler with normalized weight assignment, and periodic tasks is given by

$$W(\lceil k \rceil / k,\ \Gamma) = \frac{1-\alpha}{1/\gamma + 1} \min(1,\ k). \tag{IV-42}$$

Proof. By (IV-18), we have $\mu = 1$ and by substituting $\mu = 1$ into (IV-20) we have this corollary established. □

2. Schedulability Bound for R-Shaped Tasks

Theorem 4-9. A lower bound of schedulability bound for normalized weighted round robin scheduler with r-shaped tasks and a collection of task set Ω is given by

$$W^*(1) = \frac{1-\alpha}{1/\gamma + 1}. \qquad \text{(IV-43)}$$

Proof. By (IV-12), we know that a schedulability bound is

$$W(1,\ \Gamma) = \min_{T\in\Omega}\left(\min_{i=1,\ 2,\ \ldots,\ n}\left(\min_{I\geq 0}\left(Z(i)\right)\right)\right). \qquad \text{(IV-44)}$$

where

$$Z(i) = \max_{I\geq 0}\left(\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor \frac{TTRT-\tau}{F_i(I)} W_i(1)\right). \qquad \text{(IV-45)}$$

Rewrite (IV-45) into

$$Z(i) = (1-\alpha)\max_{I\geq 0}\left(\frac{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor}{\frac{I+D_i}{TTRT}} \frac{I+D_i}{F_i(I)} \frac{F_i(D_i)}{D_i}\right). \qquad \text{(IV-46)}$$

Since $\frac{I+D_i}{TTRT} \leq \left\lfloor \frac{I+D_i}{TTRT}\right\rfloor + 1$, we have

$$Z(i) = (1-\alpha)\max_{I\geq 0}\left(\frac{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor}{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor + 1} \frac{I+D_i}{F_i(I)} \frac{F_i(D_i)}{D_i}\right). \qquad \text{(IV-47)}$$

It can be verified that

$$\frac{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor}{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor + 1} = \frac{1}{1+\frac{1}{\left\lfloor \frac{I+D_i}{TTRT} \right\rfloor}} \geq \frac{1}{1+\frac{1}{\left\lfloor \frac{D_{\min}}{TTRT} \right\rfloor}} = \frac{1}{\gamma+1}. \tag{IV-48}$$

By substituting (IV-25) into (IV-24), we have

$$Z(i) \geq (1-\alpha)\max_{I\geq 0}\left(\frac{1}{1/\gamma+1}\frac{I+D_i}{F_i(I)}\frac{F_i(D_i)}{D_i}\right). \tag{IV-49}$$

Since F is r-shaped, we know that $F_i(D_i)/(D_i) \geq F_i(I+D_i)/(I+D_i)$ thus

$$Z(i) \geq (1-\alpha)\max_{I\geq 0}\left(\frac{1}{1/\gamma+1}\frac{F_i(I+D_i)}{F_i(I)}\right). \tag{IV-50}$$

By substituting $F_i(I+D_i) \geq F(I)$ into (IV-50) and rearrange it, we have

$$Z(i) \geq \frac{1}{1/\gamma+1}(1-\alpha). \tag{IV-51}$$

By substituting (IV-51) into (IV-44), we have

$$W^*(1) \geq \frac{1}{1/\gamma+1}(1-\alpha). \tag{IV-52}$$

That is to say, $\frac{1-\alpha}{1/\gamma+1}$ is a schedulability bound with scaling parameter $\theta = 1$. □

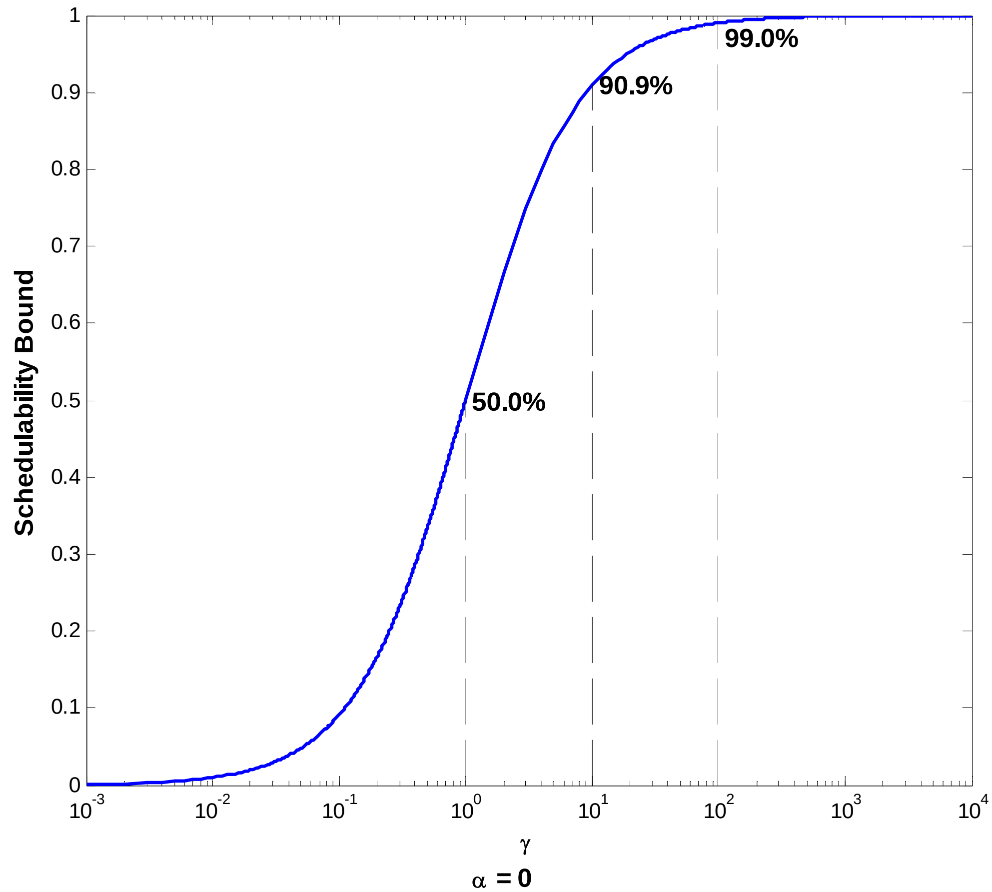

Figure 19. Schedulability Bound of Weighted Robin Scheduler with Normalized Weight Assignment and R-Shaped Tasks.

Figure 19 illustrates the relationship between the schedulability bound and the token rotation frequency. By a close observation of (IV-43) and Figure 19, we have the following conclusions:

- The schedulability bound increases with the increasing of normalized token rotation frequency γ. In other words, the faster the token rotates, the higher the schedulability bound is. When $\gamma \to \infty$, which is the GPS schedule, the schedulability bound attains its maximum 100.0%.

- The schedulability bound is not affected by the deadline assignment since there is no "deadline parameter" in (IV-43). However, this does not imply that deadline assignment has no effect on schedulability test, since the workload rate is measured in D_i window length. When D_i increases, the measured workload rate in D_i window length could decrease based on the definition of r-shaped workload constraint function (II-31).
- When the token rotates at lease once per D_{min} interval, the schedulability bound is 50.0%. This bound is the same as the schedulability bound for periodic tasks with relative deadlines equal to their periods. By a further comparison of (IV-42) and (IV-43), one can conclude that for periodic tasks with relative deadlines equal their periods ($k = 1$), the schedulability bound is same as the one for r-shaped tasks. One intuitive explanation is that weighted round robin schedulers have a "task isolation" feature compared with static priority schedulers. In a static priority scheduler, the change of high priority tasks' workload pattern will directly affect the available service for low priority ones, while in weighted round robin schedulers, each task is guaranteed to be served for its allocation per token rotation no matter how the workload pattern of other tasks change. With this "task isolation" feature, the scheduling of each individual task is similar to a single task system with slower processor in which, it is understandable that difference between periodic task model and r-shaped task model may have no

effect on schedulability bound. This is because that the single task system just needs to guarantee the total service for the task is no less than the total job arrival in D_i interval and it can be verified that when workload rate is the same, the total arrival for s-shaped task is same as r-shaped one.

C. Schedulability Bound for Deadline Based Weight Assignment

In a deadline based assignment scheme, the allocation H_i for task T_i is assigned as follows:

$$H_i = \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor}. \quad \text{(IV-53)}$$

Intuitively, this assignment scheme assigns the minimal amount of service required per round to guarantee a total service of $F_i(D_i)$ units before its deadline. Based on this assignment, it is can be verified that the token rotates at least $\lfloor D_i / TTRT \rfloor$ rounds with each round providing a service of $F_i(D_i)/\lfloor D_i / TTRT \rfloor$. This guarantees a total service of amount $F_i(D_i)$ in a window of length D_i.

Theorem 4-9. A lower bound of schedulability bound for deadline based round robin scheduler with arbitrary workload constraint function and a collection of task set Ω is given by

$$W^*(1) = \frac{1}{1/\gamma + 1}(1-\alpha), \tag{IV-54}$$

where γ is defined in (IV-3).

Proof. By (II-58) and (IV-5), we know that T_i is schedulable as long as

$$Z(i) \geq 0. \tag{IV-55}$$

where

$$Z(i) = \left\lfloor \frac{I + D_i}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor H_i - F_i(I). \tag{IV-56}$$

By substituting (IV-53) into (IV-56) and rearrange it, we get

$$Z(i) = \left\lfloor \frac{I + D_i}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} - F_i(I). \tag{IV-57}$$

Let $I = mD_i + \omega$ where m is an integer and $0 \leq \omega < D_i$. Then we have

$$Z(i) = \left\lfloor \frac{(m+1)D_i + \omega}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} - F_i\left(mD_i + \omega\right). \tag{IV-58}$$

Since F_i is non-decreasing, we have

$$Z(i) \geq \left\lfloor \frac{(m+1)D_i + \omega}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} - F_i\left((m+1)D_i\right). \tag{IV-59}$$

By property of workload constraint function, we know that

$$F_i\left((m+1)D_i\right) \le (m+1)F_i\left(D_i\right). \tag{IV-60}$$

By substituting (IV-60) into (IV-59) and rearrange it, we get

$$Z(i) \ge \left(\left\lfloor \frac{(m+1)D_i+\omega}{\sum_{i=1}^{n} H_i+\tau} \right\rfloor - (m+1)\left\lfloor \frac{D_i}{TTRT} \right\rfloor\right) \frac{F_i\left(D_i\right)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor}. \tag{IV-61}$$

It can be verified that

$$\left\lfloor \frac{(m+1)D_i+\omega}{\sum_{i=1}^{n} H_i+\tau} \right\rfloor \ge \left\lfloor \frac{(m+1)D_i}{\sum_{i=1}^{n} H_i+\tau} \right\rfloor \ge (m+1)\left\lfloor \frac{D_i}{\sum_{i=1}^{n} H_i+\tau} \right\rfloor. \tag{IV-62}$$

By substituting (IV-62) into (IV-61) and rearrange it, we have

$$Z(i) \ge \left(\left\lfloor \frac{D_i}{\sum_{i=1}^{n} H_i+\tau} \right\rfloor - \left\lfloor \frac{D_i}{TTRT} \right\rfloor\right) \frac{(m+1)F_i\left(D_i\right)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor}. \tag{IV-63}$$

By substituting (IV-63) into (IV-55), we know that T_i is guaranteed to be schedulable if

$$\left(\left\lfloor \frac{D_i}{\sum_{i=1}^{n} H_i+\tau} \right\rfloor - \left\lfloor \frac{D_i}{TTRT} \right\rfloor\right) \frac{(m+1)F_i\left(D_i\right)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} \ge 0. \tag{IV-64}$$

That is to say, T_i is guaranteed to be schedulable if

$$\sum_{i=1}^{n} H_i \le TTRT - \tau \,. \tag{IV-65}$$

By substituting (IV-53) into (IV-65), we know that T_i is guaranteed to be schedulable if

$$\sum_{i=1}^{n} \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} \le TTRT - \tau \,. \tag{IV-66}$$

An equivalent form of (IV-66) is

$$\sum_{i=1}^{n} \left(\frac{F_i(D_i)}{D_i} \frac{\frac{D_i}{TTRT}}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} \right) \le 1 - \alpha \,. \tag{IV-67}$$

Since $\frac{D_i}{TTRT} \le \left\lfloor \frac{D_i}{TTRT} \right\rfloor + 1$, we know that

$$\sum_{i=1}^{n} \left(\frac{F_i(D_i)}{D_i} \frac{\frac{D_i}{TTRT}}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} \right) \le \sum_{i=1}^{n} \left(\frac{F_i(D_i)}{D_i} \left(1 + \frac{1}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} \right) \right) \le \sum_{i=1}^{n} \left(\frac{F_i(D_i)}{D_i} \left(1 + \frac{1}{\left\lfloor \frac{D_{\min}}{TTRT} \right\rfloor} \right) \right).$$

(IV-68)

By substituting (IV-3) into (IV-68), we known that

$$\sum_{i=1}^{n} \left(\frac{F_i(D_i)}{D_i} \frac{\frac{D_i}{TTRT}}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor} \right) \le \left(\frac{1}{\gamma + 1} \right) \sum_{i=1}^{n} \left(\frac{F_i(D_i)}{D_i} \right). \tag{IV-69}$$

By substituting (IV-69) into (IV-67), we know that T_i is guaranteed to be schedulable if

$$\frac{1}{1/\gamma+1}\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\right)\leq 1-\alpha . \tag{IV-70}$$

That is to say, T_i is guaranteed to be schedulable if

$$W(1,\ \Gamma)=\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\right)\leq\frac{1}{1/\gamma+1}\left(1-\alpha\right). \tag{IV-71}$$

In other words, $W^*(1)=\dfrac{1-\alpha}{1/\gamma+1}$ is a schedulability bound with scaling parameter $\theta=1$.

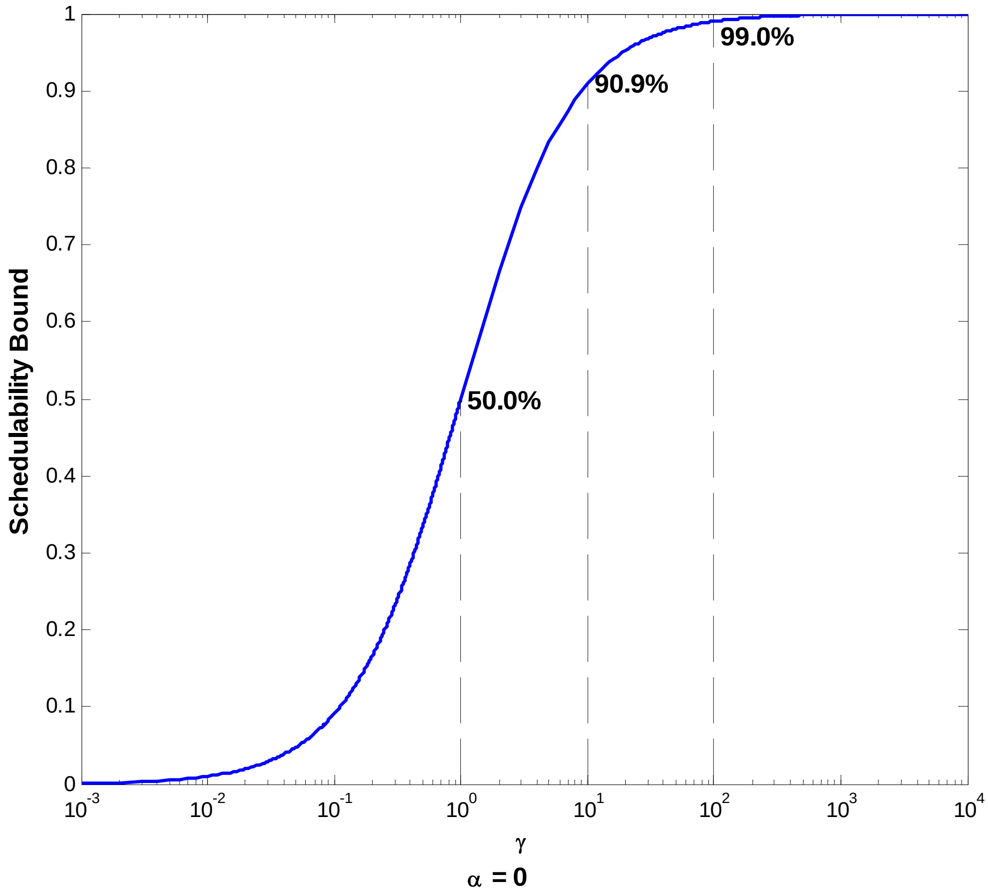

Figure 20. Schedulability Bound of Weighted Robin Scheduler with Deadline Based Weight Assignment.

Figure 20 illustrates the relationship between the schedulability bound and the normalized token rotation frequency. By a close observation of (IV-54) and Figure 20, we can draw the following conclusions:

- The schedulability bound increases with the increasing of γ. The faster the token rotates, the higher the schedulability bound is. When $\gamma \rightarrow \infty$, which is the GPS schedule, the schedulability bound attains its maximum 100.0%.

- By compare (IV-54) to (IV-43) and (IV-20), one can conclude that deadline based assignment has a schedulability bound same as normalized weight assignment with r-shaped tasks and no lower than the normalized weight assignment with s-shaped tasks. An explanation for this phenomenon (better performance of deadline based weight assignment) is that the allocation for each task is assigned in a way that each task only reserves minimum services to guarantee its own deadline. This minimum service reservation makes the scheduler serve task based on the urgent-ness of the deadline requirements and behaviors like the earliest deadline scheduler which is proven to be optimal.
- Furthermore, (IV-54) holds for arbitrary workload constraint functions, not just for s-shaped and r-shaped tasks. This generality, together with the fact that the allocation of each task is calculated only based on its own workload, makes the deadline based weight assignment scheme really simple, efficient, and flexible since the allocation of a task is not affected by the join, leave, and updates of workload of other tasks.

CHAPTER V

SCHEDULABILITY BOUND FOR TIMED TOKEN RING SCHEDULERS

A. Timed Token Ring Schedulers

In this chapter, we analyze the schedulability bound for another group of schedulers, namely the timed token ring schedulers. This type of schedulers were first proposed in [31] and have been studied extensively in [4], [5], [17], [21], [27-33], [35-38], [48], [52-55], [58-61], [63], [69-71], [75-83].

The timed token ring schedulers are similar to weighted round robin schedulers, but are different. The difference lies in the fact that each task, after receiving the token and executes H_i units of real-time jobs, may continues to execute some non-real-time jobs if the token arrives earlier in the last round. Specifically, let $TTRT$ be the target token rotation time and H_i be the allocation to task T_i. Upon receiving the token, a task can execute its real-time jobs for up to H_i time unit. H_i is calculated based on its weight factor O_i, the $TTRT$ and τ, the protocol overhead, which is the time speeded on token transversal and other protocol and network operations.

$$H_i = O_i.(TTRT - \tau) \quad \text{(V-1)}$$

After the execution of the real-time jobs, the task can continue to execution its non-real-time jobs for $TTRT - TRT$, where TRT is the actual token rotation time in the last round. More detailed description of the timed token ring scheduler can be fund in [20].

Similar to the weighted round robin scheduler, we define a parameter, normalized token rotation frequency γ for the timed token ring scheduler as

$$\gamma = \left\lfloor \frac{D_{\min}}{TTRT} \right\rfloor , \tag{V-2}$$

where $D_{\min} = \min(D_i)$ is the least relative deadline of all the tasks in the system. γ can be considered as the measured token rotation frequency in a window of length $D_{\min}$. Higher frequency leads to short scheduling delay and thus higher schedulability bound. The protocol overhead ratio α is defined as

$$\alpha = \frac{\tau}{TTRT} . \tag{V-3}$$

Theorem 5-1. For an arbitrary timed token ring scheduler, a service constraint function for task T_i, is

$$G_i(I) = \left\lfloor \frac{I}{\sum_{i=1}^{n} H_i + \tau} - 1 \right\rfloor H_i . \tag{V-4}$$

Proof. We prove (V-4) based on the definition of service constraint function. Let t be an arbitrary time instant. If at time t, all the jobs from task T_i have been served, then we can let $s = t$ and apparently (II-41) is true. Now we will focus the case that at time t, task T_i is backlogged. Let s be the last time before t such that T_i is not backlogged. That is to say, at time s, we have

$$g_i(s) = f_i(s). \quad \text{(V-5)}$$

In time interval [s, t], by property of the timed token ring scheduler, we know that the scheduler served at least $\left\lfloor \frac{I}{\sum_{i=1}^{n} H_i + \tau} - 1 \right\rfloor$ rounds with a serving time length of H_i each round. In other words, task T_i received at least $\left\lfloor \frac{I}{\sum_{i=1}^{n} H_i + \tau} - 1 \right\rfloor H_i$ amount of service time. Then we have the theorem established. □

B. Schedulability Bound for Normalized Weight Assignment

In a normalized assignment scheme, the allocation H_i is assigned as follows:

$$H_i = \frac{W_i(1)}{W(1,\ \Gamma)} \cdot \left(TTRT - \tau \right)\ . \quad \text{(V-6)}$$

Theorem 5-2. For a timed token ring scheduler with normalized weight assignment and a collection of task set Γ, a lower bound of schedulability bound is:

$$W^*(1) = \min_{T \in \Omega} \left(\min_{i=1,\ 2,\ \ldots,\ n} \left(\min_{I \geq 0} \left(\left\lfloor \frac{I + D_i}{TTRT} - 1 \right\rfloor \frac{W_i(1) \cdot \left(TTRT - \tau \right)}{F_i\left(I \right)} \right) \right) \right). \quad \text{(V-7)}$$

Proof. By (II-58), we know that for any T_i in Γ, is schedulable as long as, for any $I \geq 0$,

$$\left\lfloor \frac{I + D_i}{\sum_{i=1}^{n} H_i + \tau} - 1 \right\rfloor \frac{W_i(1) \cdot (TTRT - \tau)}{W(1,\ \Gamma)} \geq F_i(I). \tag{V-8}$$

Rewrite (IV-13) as

$$W(1,\ \Gamma) \leq \left\lfloor \frac{I + D_i}{\sum_{i=1}^{n} H_i + \tau} - 1 \right\rfloor \frac{TTRT - \tau}{F_i(I)} W_i(1). \tag{V-9}$$

It is easy to see that $\sum_{i=1}^{n} H_i + \tau = TTRT$ and

$$\left\lfloor \frac{I + D_i}{TTRT} - 1 \right\rfloor \frac{TTRT - \tau}{F_i(I)} W_i(1) \leq \min_{T \in \Omega} \left(\min_{i=1,\ 2,\ \ldots,\ n} \left(\min_{I \geq 0} \left(\left\lfloor \frac{I + D_i}{TTRT} - 1 \right\rfloor \frac{W_i(1) \cdot (TTRT - \tau)}{F_i(I)} \right) \right) \right)$$

$$. \tag{V-10}$$

By substituting (IV-15) into (IV-14), we know that T_i is schedulable as long as

$$W(1,\ \Gamma) \leq \min_{T \in \Omega} \left(\min_{i=1,\ 2,\ \ldots,\ n} \left(\min_{I \geq 0} \left(\left\lfloor \frac{I + D_i}{TTRT} - 1 \right\rfloor \frac{W_i(1) \cdot (TTRT - \tau)}{F_i(I)} \right) \right) \right). \tag{V-11}$$

Then we know the theorem is true. □

In the following subsections, we will derive schedulability bound for timed token ring schedulers with s-shaped and r-shaped workload constrain functions, respectively.

1. Schedulability Bound for S-shaped Tasks

Theorem 5-3. A lower bound of schedulability bound for timed token ring scheduler with normalized weighted assignment and a collection of task set Ω is given by

$$W\left(\lceil k \rceil / k,\ \Gamma\right) = \frac{\gamma - 1}{\gamma + 1} \cdot \left(1 - \alpha\right) \cdot \frac{1}{\mu} \cdot \min\left(1,\ k\right), \tag{V-12}$$

where γ is the normalized token rotation frequency defined in (V-2), α is the protocol overhead ratio defined in (V-3), μ is the task set burstness for s-shaped tasks defined in (IV-18), and k is normalized deadline for s-shaped tasks defined in (III-8).

Proof. By (IV-12), we know that a schedulability bound is

$$W(1,\ \Gamma) = \min_{T \in \Omega}\left(\min_{i=1,\ 2,\ \ldots,\ n}\left(\min_{I \geq 0}\left(Z(i)\right)\right)\right). \tag{V-13}$$

where

$$Z(i) = \max_{I \geq 0}\left(\left\lfloor \frac{I + D_i}{TTRT} - 1 \right\rfloor \frac{TTRT - \tau}{F_i(I)} W_i(1)\right). \tag{V-14}$$

Rewrite (IV-22) into

$$Z(i) = \max_{I \geq 0}\left(\frac{\left\lfloor \frac{I + D_i}{TTRT} \right\rfloor - 1}{\frac{I + D_i}{TTRT}} \frac{TTRT - \tau}{TTRT} \frac{I + D_i}{F_i(I)} \frac{F_i(D_i)}{D_i}\right). \tag{V-15}$$

Since $\frac{I + D_i}{TTRT} \leq \left\lfloor \frac{I + D_i}{TTRT} \right\rfloor + 1$, we have

$$Z(i) = (1-\alpha)\max_{I \ge 0}\left(\frac{\left\lfloor \frac{I + D_i}{TTRT} \right\rfloor - 1}{\left\lfloor \frac{I + D_i}{TTRT} \right\rfloor + 1} \frac{I + D_i}{F_i(I)} \frac{F_i(D_i)}{D_i} \right). \tag{V-16}$$

It is easy to verify that

$$\frac{\left\lfloor \frac{I + D_i}{TTRT} \right\rfloor - 1}{\left\lfloor \frac{I + D_i}{TTRT} \right\rfloor + 1} = 1 - \frac{2}{\left\lfloor \frac{I + D_i}{TTRT} \right\rfloor + 1} \ge 1 - \frac{2}{\left\lfloor \frac{D_{\min}}{TTRT} \right\rfloor + 1} = 1 - \frac{2}{\gamma + 1}. \tag{V-17}$$

By substituting $\frac{F_i(D_i)}{D_i} \ge \frac{F_i(D_i)}{I + D_i}$ into (V-17) and rearrange it, we have

$$Z(i) \ge (1-\alpha)\frac{\gamma - 1}{\gamma + 1}\max_{I \ge 0}\left(\frac{I + D_i}{F_i(I)} \frac{F_i(D_i)}{D_i} \right). \tag{V-18}$$

Now let $I = mS_i + \omega$ where 0≤$0 \le \omega \le S_i$ <Si. By (II-11), we have

$$F_i(I) \le F_i((m+1)S_i). \tag{V-19}$$

By substituting (V-19) into (V-18) and rearrange it, we have

$$Z(i) \ge (1-\alpha)\frac{\gamma - 1}{\gamma + 1}\max_{I \ge 0}\left(\frac{mS_i + \omega + D_i}{F_i((m+1)S_i)} \frac{F_i(D_i)}{D_i} \right). \tag{V-20}$$

Since $0 \le \omega$, we have

$$Z(i) \ge (1-\alpha)\frac{\gamma - 1}{\gamma + 1}\max_{I \ge 0}\left(\frac{mS_i + D_i}{F_i((m+1)S_i)} \frac{F_i(D_i)}{D_i} \right). \tag{V-21}$$

By (II-15) we have

$$\frac{F_i\left((m+1)S_i\right)}{(m+1)S_i} \le \frac{F_i\left(S_i\right)}{S_i}. \tag{V-22}$$

By substituting (V-22) into (V-21) and rearrange it, we get

$$Z(i) \ge \left(1-\alpha\right)\frac{\gamma-1}{\gamma+1}\max_{I\ge 0}\left(\frac{mS_i + D_i}{(m+1)F_i\left(S_i\right)}\frac{F_i(D_i)}{D_i}\right). \tag{V-23}$$

By definition of s-shaped workload constraint function (II-11), we know that

$$F_i(D_i) = F_i(kS_i) = F_i(\lceil k \rceil S_i). \tag{V-24}$$

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

$$Z(i) \ge \left(1-\alpha\right)\frac{\gamma-1}{\gamma+1}\max_{I\ge 0}\left(\frac{mS_i + D_i}{(m+1)F_i\left(S_i\right)}\frac{F_i(\lceil k \rceil S_i)}{kS_i}\right). \tag{V-25}$$

Rewrite (V-25) into

$$Z(i) \ge \left(1-\alpha\right)\frac{1}{k}\frac{\gamma-1}{\gamma+1}\frac{F_i(\lceil k \rceil D_i)}{F_i\left(S_i\right)}\max_{m=0,\ 1,\ 2,\ \ldots}\left(\frac{m+k}{m+1}\right). \tag{V-26}$$

It is easy to verify that

$$\max_{m=0,\ 1,\ 2,\ \ldots}\left(\frac{m+k}{m+1}\right) \ge \min\left(1,\ k\right). \tag{V-27}$$

By substituting (V-27) into (V-26), we have

$$Z(i) \geq \frac{1}{k}\frac{\gamma-1}{\gamma+1}\frac{F_i(\lceil k \rceil S_i)}{F_i\left(S_i\right)}\left(1-\alpha\right)\min\left(1,\ k\right). \tag{V-28}$$

By (IV-18), we have

$$\frac{F(\lceil k \rceil S_i)}{F(S_i)} = \frac{\lceil k \rceil}{\mu_i} \geq \frac{\lceil k \rceil}{\mu} \ . \tag{V-29}$$

By substituting (V-29) into (V-28), we get

$$Z(i) \geq \frac{\gamma-1}{\gamma+1}\frac{\lceil k \rceil}{k\mu}\left(1-\alpha\right)\min\left(1,\ k\right). \tag{V-30}$$

By substituting (V-30) into (V-29), we have a schedulability bound of

$$W\left(1,\ \Gamma\right) = \frac{\gamma-1}{\gamma+1}\frac{\lceil k \rceil}{k\mu}\left(1-\alpha\right)\min\left(1,\ k\right). \tag{V-31}$$

By (II-11), we know that

$$W\left(1,\ \Gamma\right) = \sum\nolimits_{i=1}^{n}\frac{F_i(D_i)}{D_i} = \frac{\lceil k \rceil}{k}\sum\nolimits_{i=1}^{n}\frac{F_i(\lceil k \rceil S_i)}{\lceil k \rceil S_i} = \frac{\lceil k \rceil}{k}\sum\nolimits_{i=1}^{n}W\left(\frac{\lceil k \rceil}{k},\ \Gamma\right). \tag{V-32}$$

By substituting (V-32) into (V-31), we have a schedulability bound with scaling parameter $\theta = \lceil k \rceil / k$ in form of

$$W\left(\lceil k \rceil / k,\ \Gamma\right) = \frac{\gamma-1}{\gamma+1}\cdot\left(1-\alpha\right)\cdot\frac{1}{\mu}\cdot\min\left(1,\ k\right). \tag{V-33}$$

Then we have the theorem proven. □

By a close observation of (V-12), one can notice:

- Given protocol overhead ratio α, normalized deadline k, and task set workload burstness μ, higher token rotation frequency leads to improved schedulability bound. The schedulability bound achieves its highest value when $\gamma \to \infty$, when the scheduler is GPS. When $\gamma < 1$, the schedulability bound reduces to zero. That is to say, when it takes longer than the minimum relative deadline to finish one round of token rotation, the schedulability bound is zero. This is understandable since a task may not receive the token within its deadline and thus may not be able to receive any service at all. Figure 21 illustrates this trend for the special case k =1 and $\alpha = 0$.
- Given protocol overhead ratio α, normalized deadline k, and token rotation frequency γ, increasing of task set workload burstness μ results in decreased schedulability bound. This is because the larger the μ, the bursty the workload is. Typicaly, bursty workloads are more difficult schedule than less burstness ones. The schedulability bound is maximized when μ =1 for periodic tasks. Figure 22 illustrates this trend for the case of k =1. Note that the workload rate is measured in the window length of $D_i \lceil k_i \rceil / k_i$, not S_i in (V-12).
- Given protocol overhead ratio α, token rotation frequency γ, and task set workload burstness μ, relaxed deadline requirements (larger normalized deadline k) leads to improved schedulability bound. When k <1, the schedulability bound

increases linearly with the increasing of k. When $k>1$, the schedulability bound does not change with the increase of k. This trend is shown as Figure 23. We should note that this does not imply that, when $k>1$, relaxing deadline requirement has no effect on schedulability test, because, according to (IV-10) and (II-12), the calculated workload rate in the window D_i may decrease with the increasing of k. This decreasing of workload rate is equivalent to the increasing of schedulability bound.

- For periodic tasks, when $k=1$, $\mu=1$, and $\gamma=2$ (token rotates twice per round), our newly derived bound reduces to 33.3% which is first derived in [5] and is corresponding to a point on the 3-D surface in Figure 21.
- By a close comparison between (IV-20) and (V-12), we notice that the schedulability bound for time token ring scheduler is lower than weighted round robin schedulers. For example, when when $k=1$, $\mu=1$, and $\gamma=2$, the schedulability bound for weighted round robin scheduler is 66.7%, but only 33.3% for timed token ring scheduler. This is due to the interference of non real-time jobs. However, the difference between the two bounds gradually reduces with the increasing of token rotation frequency and approaches zero when $\gamma\rightarrow\infty$ for GPS scheduler.

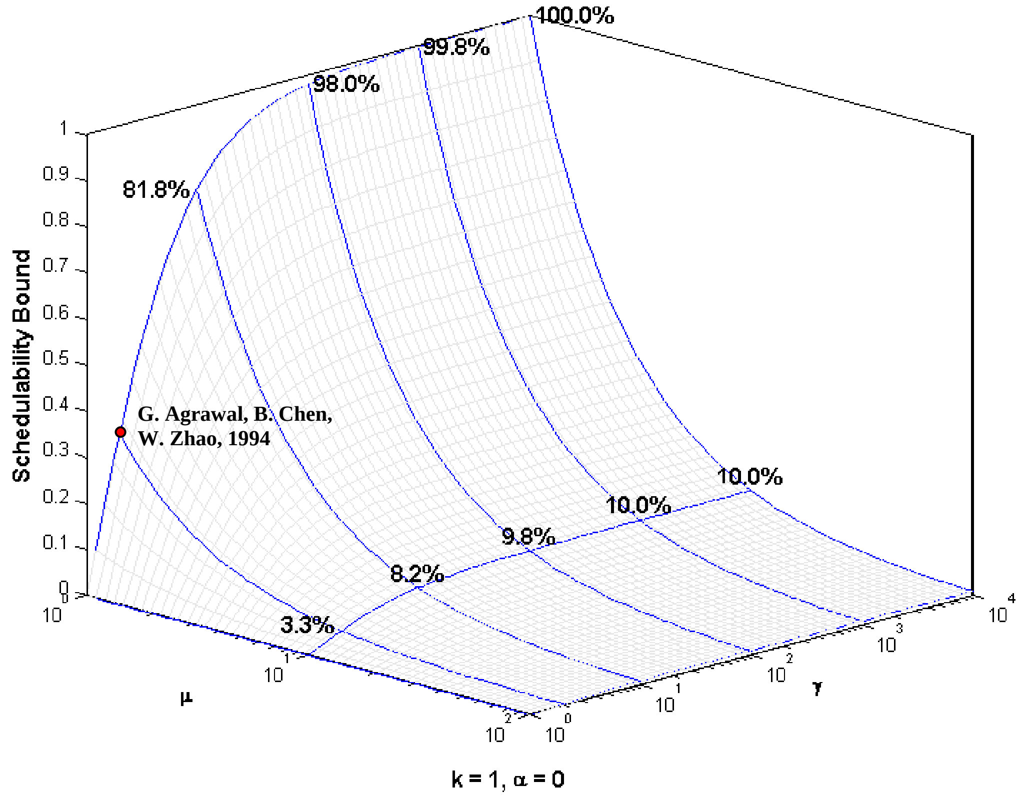

Figure 21. Schedulability Bound of Timed Token Ring Scheduler with Normalized Weight Assignment for and Fixed Normalized Deadline $k = 1$.

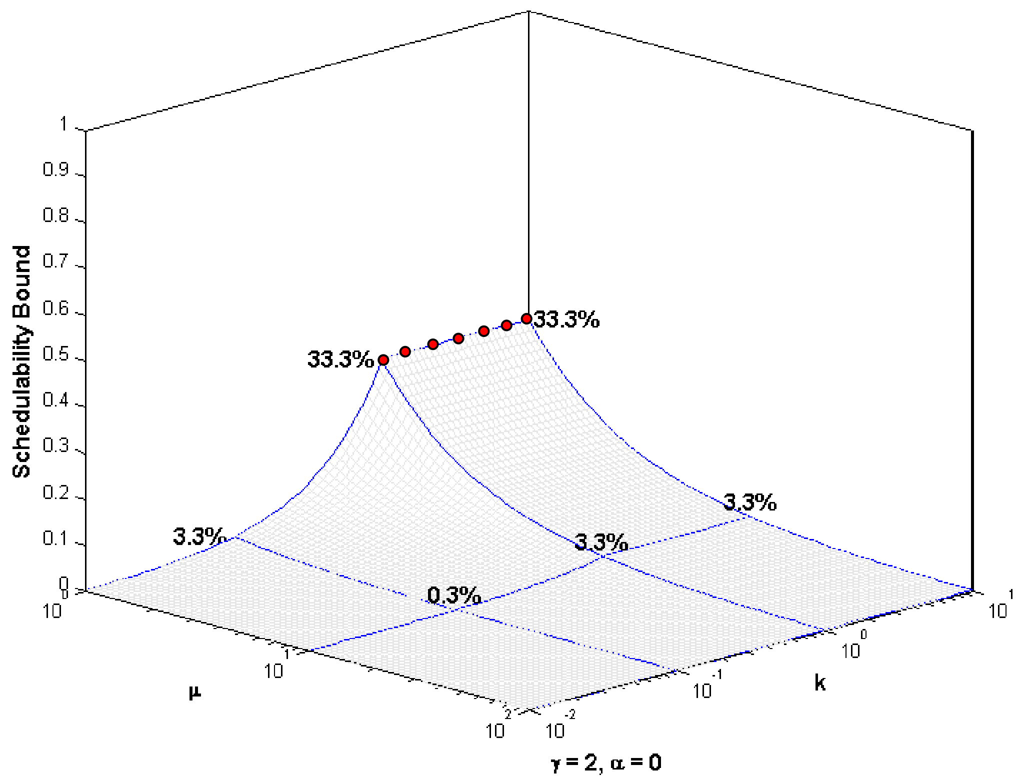

Figure 22. Schedulability Bound of Timed Token Ring Scheduler with Normalized Weight Assignment and Fixed Token Rotation Frequency γ=2.

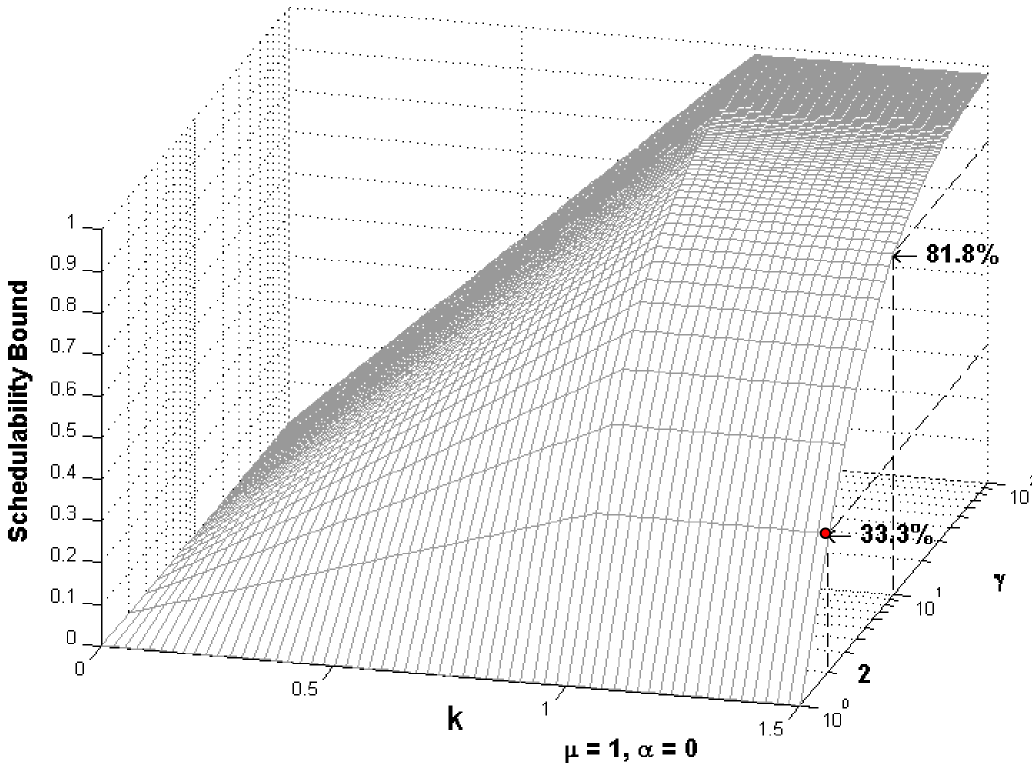

Figure 23. Schedulability Bound of Timed Token Ring Scheduler with Normalized Weight Assignment and Fixed Workload Burstness μ =1.

2. Schedulability Bound for R-shaped Tasks

Theorem 5-4. A lower bound of schedulability bound for timed token ring scheduler and a collection of task r-shaped set Ω is given by

$$W^*(1) \geq \frac{\gamma - 1}{\gamma + 1} \cdot (1 - \alpha). \qquad \text{(V-34)}$$

Proof. By (IV-12), we know that a schedulability bound is

$$W(1,\ \Gamma) = \min_{T\in\Omega}\left(\min_{i=1,\ 2,\ \dots,\ n}\left(\min_{I\geq 0}\left(Z(i)\right)\right)\right). \tag{V-35}$$

where

$$Z(i) = \max_{I\geq 0}\left(\left\lfloor \frac{I+D_i}{TTRT} - 1 \right\rfloor \frac{TTRT-\tau}{F_i(I)} W_i(1)\right). \tag{V-36}$$

Rewrite (V-36) into

$$Z(i) = \max_{I\geq 0}\left(\frac{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor - 1}{\frac{I+D_i}{TTRT}}(1-\alpha)\frac{I+D_i}{F_i(I)}\frac{F_i(D_i)}{D_i}\right). \tag{V-37}$$

Since $\frac{I+D_i}{TTRT} \leq \left\lfloor \frac{I+D_i}{TTRT}\right\rfloor + 1$, we have

$$Z(i) = \max_{I\geq 0}\left(\frac{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor - 1}{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor + 1}(1-\alpha)\frac{I+D_i}{F_i(I)}\frac{F_i(D_i)}{D_i}\right). \tag{V-38}$$

It is easy to verify that

$$\frac{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor - 1}{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor + 1} = 1 - \frac{2}{\left\lfloor \frac{I+D_i}{TTRT}\right\rfloor + 1} \geq 1 - \frac{2}{\left\lfloor \frac{D_{\min}}{TTRT}\right\rfloor + 1} = 1 - \frac{2}{\gamma+1} = \frac{\gamma-1}{\gamma+1}. \tag{V-39}$$

By substituting (V-39) into (V-38), we have

$$Z(i) \geq (1-\alpha)\max_{I\geq 0}\left(\frac{\gamma-1}{\gamma+1}\frac{I+D_i}{F_i(I)}\frac{F_i(D_i)}{D_i}\right). \tag{V-40}$$

Since F is r-shaped, we know that $F_i(D_i)/(D_i) \geq F_i(I+D_i)/(I+D_i)$ thus

$$Z(i) \geq (1-\alpha)\max_{I\geq 0}\left(\frac{\gamma-1}{\gamma+1}\frac{F_i(I+D_i)}{F_i(I)}\right). \tag{V-41}$$

By substituting $F_i(I+D_i) \geq F(I)$ into (V-41) and rearrange it, we have

$$Z(i) \geq (1-\alpha)\frac{\gamma-1}{\gamma+1}. \tag{V-42}$$

By substituting (V-42) into (V-35), we have

$$W^*(1) \geq (1-\alpha)\frac{\gamma-1}{\gamma+1}. \tag{V-43}$$

That is to say, $\frac{\gamma-1}{\gamma+1}(1-\alpha)$ is a schedulability bound with scaling parameter $\theta = 1$. □

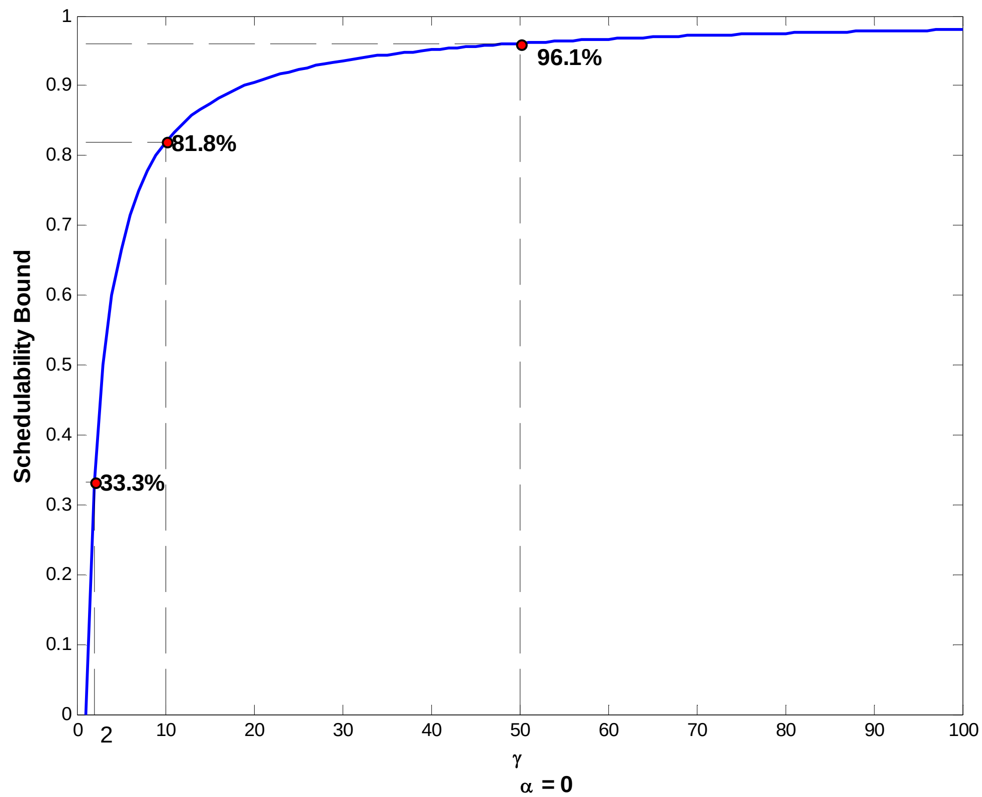

Figure 24. Schedulability Bound of Timed Token Ring Scheduler with Normalized Weight Assignment and R-Shaped Tasks.

Figure 24. illustrates the relationship between the schedulability bound and the token rotation frequency. By a close observation of (V-34) and Figure 24, we can have the following conclusions:

- The schedulability bound increases with the increasing of token rotation frequency γ. The higher the frequency, the greater the schedulability bound is. When $\gamma \to \infty$, which is the GPS schedule, the schedulability bound attains its maximum 100.0%.

- The sensitivity of the schedulability bound on token ration rotation frequency γ is high for small γ, e.g. $\gamma < 10$, and gradually reduces. For example, when the frequency increases from 2 to 10, the schedulability increases from 33.3% to 81.8%, a 48.5% gain. When the frequency increases from 10 to 50, the bound changes from 81.8% to 96.1%, a mere 14.3% gain.
- By a close comparison between (V-34) and (IV-43), we notice that the schedulability bound for time token ring scheduler is lower than weighted round robin schedulers. For example, when $\gamma = 2$, the schedulability bound for weighted round robin scheduler is 66.7%, but only 33.3% for timed token ring scheduler. This is due to the interference of non real-time jobs. However, the difference between the two bounds gradually reduces with the increasing of token rotation frequency and approaches zero when $\gamma \to \infty$ for GPS scheduler.

C. Schedulability Bound for Deadline Based Weight Assignment

In a deadline based assignment scheme, the allocation H_i is assigned as follows:

$$H_i = \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} - 1 \right\rfloor}, \tag{V-44}$$

where for $i = 1,\ 2,\ \ldots,\ n$.

Theorem 5-5. A lower bound of schedulability bound for deadline based round robin scheduler with arbitrary workload constraint function and a collection of task set Ω is given by

$$W^*(1) = \frac{\gamma - 1}{\gamma + 1}\left(1 - \alpha\right), \qquad \text{(V-45)}$$

where γ is defined in (IV-3).

Proof. By (II-58) and (IV-5), we know that T_i is schedulable as long as

$$Z(i) \geq 0. \qquad \text{(V-46)}$$

where

$$Z(i) = \left\lfloor \frac{I + D_i}{\sum_{i=1}^{n} H_i + \tau} - 1 \right\rfloor H_i - F_i(I). \qquad \text{(V-47)}$$

By substituting (V-44) into (V-47) and rearrange it, we get

$$Z(i) = \left\lfloor \frac{I + D_i}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1} - F_i(I). \qquad \text{(V-48)}$$

Let $I = mD_i + \omega$ where m is an integer and $0 \leq \omega < D_i$. Then we have

$$Z(i) = \left\lfloor \frac{(m+1)D_i + \omega}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1} - F_i\left(mD_i + \omega\right). \qquad \text{(V-49)}$$

Since F_i is non-decreasing, we have

$$Z(i) \geq \left\lfloor \frac{(m+1)D_i + \omega}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor \frac{F_i(D_i)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1} - F_i\left((m+1)D_i\right). \tag{V-50}$$

By the triangle-property of workload constraint function, we know that

$$F_i\left((m+1)D_i\right) \leq (m+1)F_i\left(D_i\right). \tag{V-51}$$

By substituting (V-51) into (V-50) and rearrange it, we get

$$Z(i) \geq \left(\left\lfloor \frac{(m+1)D_i + \omega}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor - (m+1) \left\lfloor \frac{D_i}{TTRT} \right\rfloor \right) \frac{F_i\left(D_i\right)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1}. \tag{V-52}$$

It is easy to verify that

$$\left\lfloor \frac{(m+1)D_i + \omega}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor \geq \left\lfloor \frac{(m+1)D_i}{\sum_{i=1}^{n} H_i} \right\rfloor \geq (m+1) \left\lfloor \frac{D_i}{\sum_{i=1}^{n} H_i} \right\rfloor. \tag{V-53}$$

By substituting (V-53) into (V-52) and rearrange it, we have

$$Z(i) \geq \left(\left\lfloor \frac{D_i}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor - \left\lfloor \frac{D_i}{TTRT} \right\rfloor \right) \frac{(m+1)F_i\left(D_i\right)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1}. \tag{V-54}$$

By substituting (V-54) into (V-46), we know that T_i is guaranteed to be schedulable if

$$\left(\left\lfloor \frac{D_i}{\sum_{i=1}^{n} H_i + \tau} \right\rfloor - \left\lfloor \frac{D_i}{TTRT} \right\rfloor\right) \frac{(m+1)F_i\left(D_i\right)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1} \geq 0 . \quad \text{(V-55)}$$

That is to say, T_i is guaranteed to be schedulable if

$$\sum_{i=1}^{n} H_i \leq TTRT - \tau . \quad \text{(V-56)}$$

By substituting (V-44) into (V-56), we know that T_i is guaranteed to be schedulable if

$$\sum_{i=1}^{n} \frac{F_i\left(D_i\right)}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1} \leq TTRT - \tau . \quad \text{(V-57)}$$

An equivalent form of (V-57) is

$$\sum_{i=1}^{n} \left(\frac{F_i\left(D_i\right)}{D_i} \frac{\frac{D_i}{TTRT - \tau}}{\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1} \right) \leq 1 . \quad \text{(V-58)}$$

Since $\frac{D_i}{TTRT} \leq \left\lfloor \frac{D_i}{TTRT} \right\rfloor + 1$, we know that

$$\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\frac{\frac{D_i}{TTRT-\tau}}{\left\lfloor\frac{D_i}{TTRT}\right\rfloor-1}\right)\le\left(1-\alpha\right)\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\left(\frac{\left\lfloor\frac{D_i}{TTRT}\right\rfloor+1}{\left\lfloor\frac{D_i}{TTRT}\right\rfloor-1}\right)\right). \qquad \text{(V-59)}$$

$$\le\left(1-\alpha\right)\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\left(1-\frac{\gamma+1}{\gamma-1}\right)\right)$$

By substituting (V-59) into (V-58), we known that

$$\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\frac{\frac{D_i}{TTRT}}{\left\lfloor\frac{D_i}{TTRT}\right\rfloor}\right)\le\left(1-\alpha\right)\frac{\gamma+1}{\gamma-1}\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\right). \qquad \text{(V-60)}$$

By substituting (V-60) into (V-58), we know that T_i is guaranteed to be schedulable if

$$\left(1-\alpha\right)\frac{\gamma+1}{\gamma-1}\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\right)\le 1. \qquad \text{(V-61)}$$

That is to say, T_i is guaranteed to be schedulable if

$$W(1,\ \Gamma)=\sum_{i=1}^{n}\left(\frac{F_i\left(D_i\right)}{D_i}\right)\le\frac{\gamma-1}{\gamma+1}\left(1-\alpha\right). \qquad \text{(V-62)}$$

In other words, $W^*(1)=\frac{1-\gamma}{1+\gamma}\left(1-\alpha\right)$ is a schedulability bound with scaling parameter $\theta=1$. □

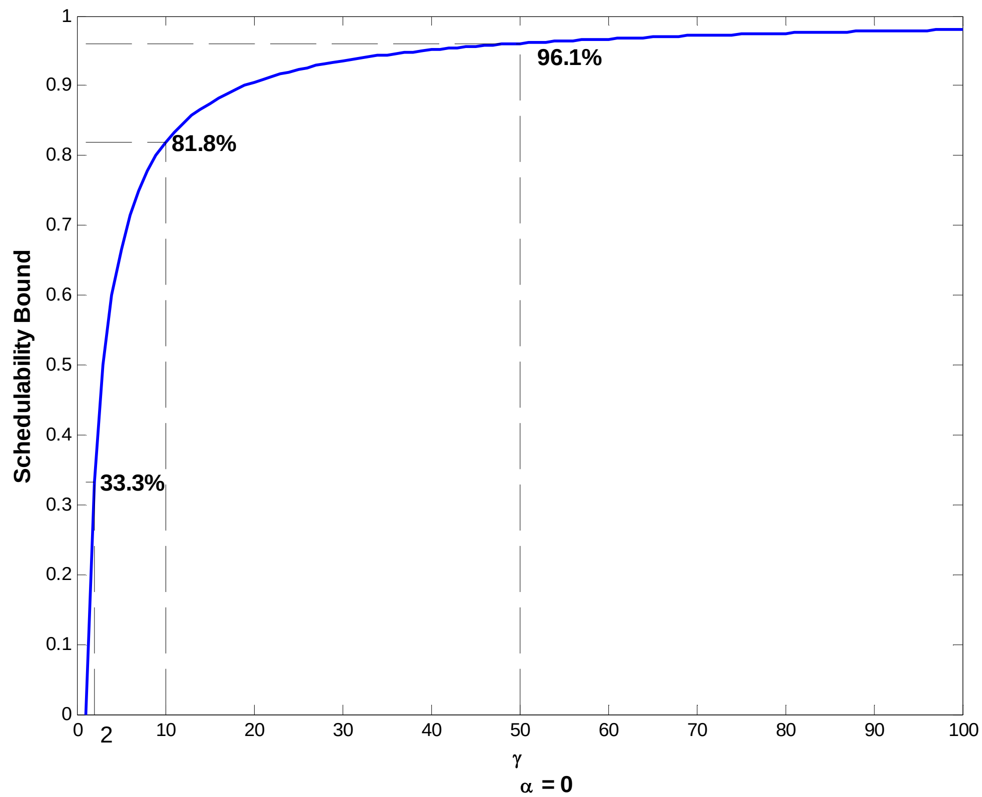

Figure 25. Schedulability Bound of Timed Token Ring Scheduler with Deadline Based Weight Assignment.

Figure 25 illustrates the relationship between the schedulability bound and the token rotation frequency. By a close observation of (V-45) and Figure 25, we have the following observations:

- The schedulability bound increases with the increasing of γ. The faster the token rotates, the higher the schedulability bound is. When $\gamma \to \infty$, which is the GPS schedule, the schedulability bound attains its maximum 100.0%.
- By comparing (V-45) to (V-12) and (V-34), one can conclude that deadline based assignment has a schedulability bound same as normalized weight assignment

with r-shaped tasks and no lower than the normalized weight assignment with s-shaped tasks. An explanation for this phenomenon (better performance of deadline based weight assignment) is that the allocation for each task is assigned in a way that each task only reserves minimum services to guarantee its own deadline. This minimum service reservation makes the scheduler serve task based on the urgent-ness of the deadline requirements and behaviors closers to earliest deadline scheduler which is proven to be optimal.

- Furthermore, (V-45) holds for arbitrary workload constraint functions, not just for s-shaped and r-shaped tasks. This generality, together with the fact that the allocation of each task is calculated only based on its own workload, makes the deadline based weight assignment scheme really simple, efficient, and flexible since the allocation of a task is not affected by the join, leave, and updates of workload of other tasks.
- In [4], [53], [83], the authors derived a similar schedulability bound for periodic tasks which is a special case of (V-45) since (V-45) holds for arbitrary workload constraint functions.

CHAPTER VI

APPLICATIONS

A. Background

In the past years, Texas A&M University has been involved in a large number of scientific modeling and data analysis projects. These projects share a data archive and retrieval requirement, a need to visualize data, a call to perform customized modeling tasks using complex environmental, atmospheric, oceanographic or geophysical models, and a need to display data in a geospatially referenced manner. Most of these project have high demands on computing resources and the model runs must be finished timely so that the result to be of any use, e.g. hurricane forecasts. Many other projects at different research institutions and organizations share similar needs. Typically, the institutions and organizations will design and procure hardware and software to address these problems for each individual project in an ad hoc manner. The high implementation cost, the duplication of efforts, and the difficulty of collaboration due to the lack of standardization for working process and data format, often leave something to be desired.

The Reference Center for Modeling and Data Analysis (RCMDA) is established to address all of these issues and will, further provide a reference case for implementation of the Data Center model by creating and operating a working data center, and addressing all identified tasks and requirements to achieve full functionality while providing a working example and demonstration of the various aspects of such a project.

B. System Architecture

Figure 26 illustrates the system architecture. On the left-most side is the user interface layer which includes three major object groups a typical user can manipulates: workflows, tasks, and admitted tasks. A workflow is composed of a set of inter-related models. Each model is an executable with one or more inputs and produces one or more outputs. The model could be a scientific simulation program, a data retrieval program, a visualization package, or any other computing process. The outputs of one model could be the input of the model. A workflow specification defines the relationship between the models, e.g. input/output relationship, execution sequence, resource requirement, and some other workflow parameters, e.g. time interval of a simulation or simulation length. Once the workflow parameters are filled, a workflow becomes a task and is ready to be executed on the clusters. A user can send an admit request to the system to run the task on the cluster. If the request is granted, the task will start execution and we say the task is an admitted task.

On the right-most side are the cluster resources which are categorized into different groups based their functionalities: computing cluster, data processing cluster, and graphic cluster. The computing cluster is used for scientific computing; the data processing cluster is used for data retrieving and sharing, e.g. download data from external web site and upload simulation output to collaborators' file server; the graphic cluster is used to generate graphics or supporting interactive visualization of the simulation data. Separation of the clusters is necessary since different clusters have different capabilities, e.g. graphic cluster must have special graphic packages installed, and the nodes in data processing cluster must be able to connect to the Internet.

In the middle are the system components including admission control, job generator, scheduler, and job dispatcher. Admission control is used to check the schedulability of the tasks. Upon the admit request of a task, the admission control checks whether the new task and the admitted tasks can meet their deadlines. If yes, the admission control will return success and the task will be added to the admitted task table. If not, the admission control returns failure and the task is rejected. For each admitted job, a job generator instance is created and this instance is responsible for generation of the individual jobs for this task based on its task specifications, e.g. execution periods. All the generated jobs are appended to the incoming job queue which is managed by the scheduler. The scheduler picks the jobs in the job queue one by one and inserts them into the outgoing job queue. The jobs in the outgoing job queue are arranged based on the scheduling policy, e.g. high priority job is in the front of the queue. The job dispatcher is responsible for monitoring the cluster and picking the jobs from the outgoing job queue and dispatching them to the cluster once the cluster is free. The job dispatcher also monitors the progress of the jobs running on the cluster and notifies the scheduler, admission control, job generator once a job is finished.

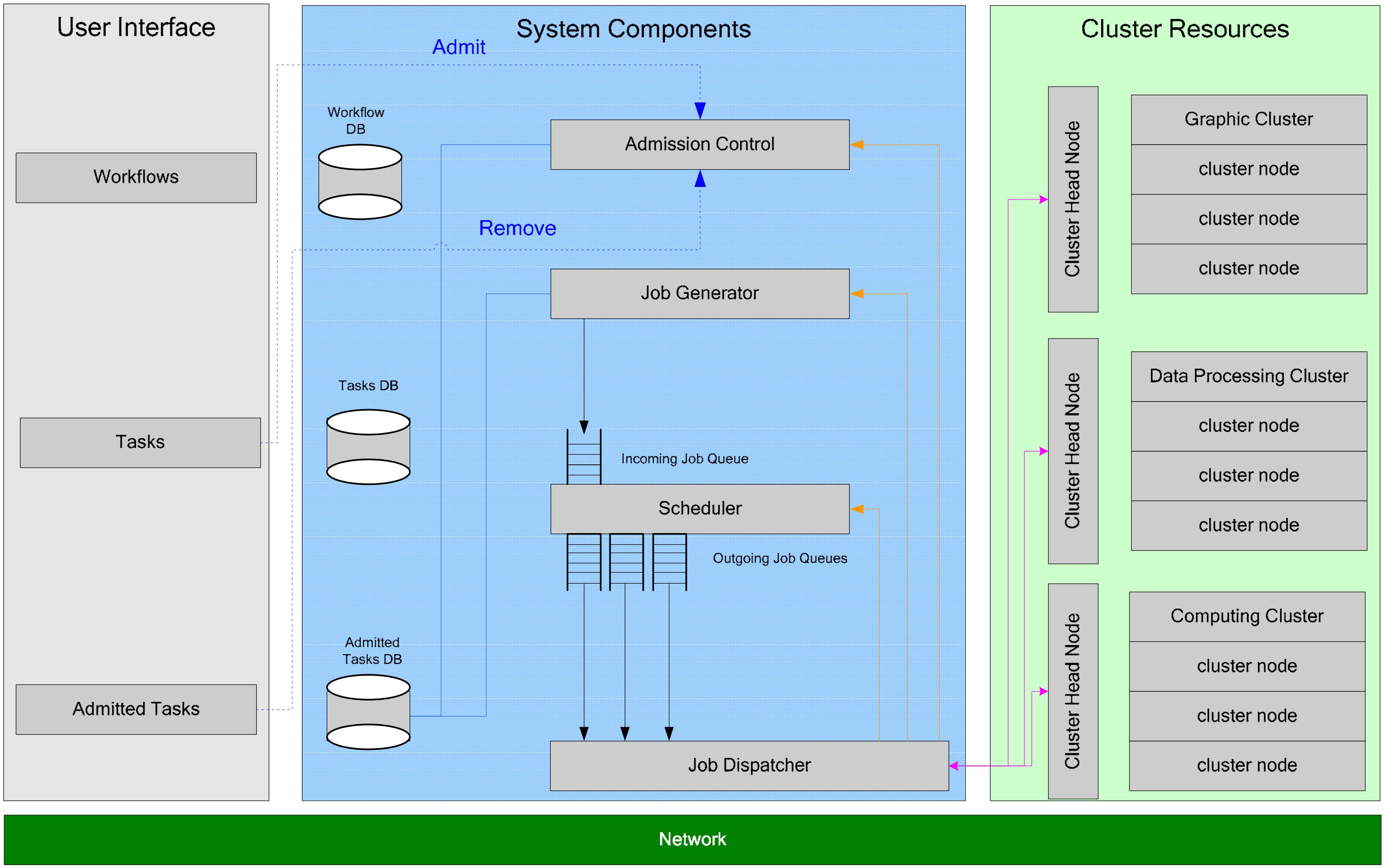

Figure 26. System Architecture of the RCMDA.

C. Implementation

The system is implemented in a LINUX environment using Java/J2EE as the main programming language. The clusters are managed by the PBS (Portable Batch Scheduling System). The clusters connected with gigabit switch. In the following sections, we will discuss some of the implementation details.

1. Tasks Modeling

In this implementation, we use a special r-shaped workload constraint function to model the task workload. We selected leaky bucket workload constraint function for its extreme simplicity. For each task, we collect two workload parameters: σ and ρ, where σ is the burstness and ρ is the long-term rate. For each task, we also collect its relative deadline D. The relative deadline is specified by the end user when create a task.

Recall that a task is a workflow with input parameter set and a workflow is composed of a sequence of models. The execution times of the models are obtained by pre-run these models in a standard cluster. The execution times of each workflow are then calculated automatically from these model execution times. During admission time, the execution time is then normalized to the real cluster environment. The normalization function is obtained using heuristic approach. The two workload constraint function parameters: σ and ρ are then calculated based on this execution time of each workflow and the execution patterns of the task specified by the end user, e.g. run every Friday, or no more than twice per week and at lease one day apart between two consecutive runs.

2. Scheduler

We use static priority scheduler in current implementation. The scheduler maintains a job queue for each cluster and the jobs in the queue are arranged based on their priorities. That is, the jobs from the task with high priority are inserted in front of the low priority ones. After admission, each task is assigned to a specific cluster for execution. For each new job, the scheduler finds the cluster on which it will run and insert the job to the outgoing job queue corresponding to the cluster. Since the queue is ordered based on their priorities, the insert operations is very efficient using the binary search algorithm with a complexity of O(log(n)).

3. Admission Control

We use schedulability bound based admission control in this implementation. Recall that a set of r-shaped tasks is schedulable if

$$\sum_{i=1}^{n} F_i(D_i)/D_i < 1/\lambda, \tag{VI-1}$$

where λ is the deadline inversion and is calculated as

$$\lambda = \max_{i=1,\ 2,\ \ldots,\ n}\left(\frac{\max_{j\le i}(D_j)}{D_i}\right). \tag{VI-2}$$

For leaky bucket tasks, an equivalent form of (VI-1) is,

$$\sum_{i=1}^{n} \frac{\sigma_i + \rho_i D_i}{D_i - J_{\max}} < 1/\lambda, \tag{VI-3}$$

where J_{max} is the maximum job size.

To perform the test, we need to calculate the deadline inversion ratio λ. Though we can calculate λ upon the join request of a new task using (VI-2), the complexity of this calculation is O(n^2). To reduce the complexity, we introduce a different deadline inversion ratio calculation algorithm described in Figure 27.

```
TaskList : An array of the tasks currently admitted to the system. The array is sorted based on the
           relative deadlines in ascending order.
n: Number of tasks in the task list.
CurrentDeadlineInversion: The current deadline inversion ratio in the system.
T: The task to be admitted.

i  =  the last task in the task array whose relative deadline is no more than the one of task T.
//Fist, let us find the deadline inversion ratio for task T if it is admitted
// Note that the deadline inversion for any existing task with higher priority than T will not be affected
For j = 0 to i-1
      Tj = the jth task in the TaskList
      ThisInversion = Relative Deadline of Tj/Relative Deadline of T
       If  ThisInversion > CurrentDeadlineInversion Then
              CurrentDeadlineInversion = ThisInversion
      End if
Next
// Now calculate the maximum deadline inversion for low priority tasks.
For j = i to n
      Tj = the jth task in the TaskList
      ThisInversion = Relative Deadline of T/  Relative Deadline of Tj
       If  ThisInversion > CurrentDeadlineInversion Then
              CurrentDeadlineInversion = ThisInversion
      End if
Next
return CurrentDeadlineInversion;
```

Figure 27. Pseudo-code to Update Degree of Deadline Inversion.

One can verify that the above algorithm has a complexity of O(*n*) which is much more efficient then the calculation of deadline inversion directly using (VI-2).

Another important functionality of the admission control component is handling the task departure, e.g. deleted tasks, or finished tasks. The utilization reserved for these deleted tasks must be recycled. In our current implementation, this recycle will not happen until the system is idle. The reason for this is two folds: one is it has been proved that the reserved resource of a denatured task can not be recycled at the time of departure since the task may already used the resource, and second, recycling the reserved resource at the system idle time is more efficient and safe since the system can be treated as a "restart".

4. Job Generator

The job generator is an active program that monitors the execution progress of the jobs and release next job based on the task execution pattern specified by the user during the admission time, e.g. execute the job every Friday. The job generator is created at the time a new task is admitted and is destroyed after the task is finished or removed. The generated jobs are sent to the incoming job queue of the scheduler. Most of the time, the job generator will be in sleep stage and is waken up by the job dispatcher when a job is finished normally or terminated inexpertly. Once awaken, the job scheduler parses the task specification and decide which job should be released next.

5. Job Dispatcher

The job dispatcher is implemented as an active service that periodically checks the cluster status. If the job dispatcher finds that the cluster is free (empty or partially empty), it will perform the following two operations:

- Notify the admission control, scheduler, and the job generator.
- Get the next job from the scheduler.

Note that the job dispatcher may not be able to get a job from the scheduler. If this is the case, the job dispatcher will change to sleep mode until the next polling time.

CHAPTER VII

SUMMARY AND CONCLUSIONS

This dissertation addresses the following problem:

How to derive schedulability bound for general real-time systems?

Based on network calculus theory, we proposed a general schedulability bound analysis framework for real-time systems. The general framework uses workload constraint function to model tasks, service constraint function for schedulers, and workload rate for utilization measurement. We proposed two new special forms of workload constraint functions that are flexible and accurate in modeling task behavior. To show the powerfulness of the framework, we derived closed-form schedulability bound for arbitrary static priority schedulers, weighted round robin schedulers, and timed token ring schedulers. The bounds are parameterized for different system configuration. By simple plug-in of parameters, we show that most of the existing bound results are special cases of our new generalized bound results. We also applied some of the result in a real world cluster computing project.

This work has several limitations and can be improved in many directions. Only single processor systems are considered in this dissertation. Extension of the framework and schedulability analysis to multi-processor system will be an interesting and challenging task. Another direction of extension is to the networked computing environment in which a job may take several hop to finish, e.g. delivering package from one node to another node through a set of intermediate routers. Additional type workload constraint functions can also be proposed. Though we implemented the schedulability bound based admission control and a static priority scheduler in a cluster computing environment, the modeling of the task workloads is not accurate and can be improved by future research.

REFERENCES

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

[2] T. Abdelzaher and V. Sharma, "A synthetic utilization bound for aperiodic tasks with resource requirements," *Proc. 15th Euromicro Conference on Real-Time Systems*, pp. 141-150, Jul. 2003.

[3] T. Abdelzaher, V. Sharma, and C. Lu, "A utilization bound for aperiodic tasks and priority driven scheduling," *IEEE Trans. Computers*, vol. 53, no. 3, pp. 334-350, Mar. 2004

[4] G. Agrawal, B. Chen, and W. Zhao, "Local synchronous capacity allocation schemes for guaranteeing message deadlines with the timed token protocol," *Proc. IEEE INFOCOM'93*, pp. 186-193, Mar. 1993.

[5] 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.

[6] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan, "A framework for adaptive service guarantees," *Proc. 36th Allerton Conference on Communication, Control, and Computation*, pp. 693-702, Sept. 1998.

[7] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan, "Performance bounds for flow control protocols," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp 310-323, Jun. 1999.

[8] R. Amitava, N. Malcolm, and W. Zhao, "Hard real-time communications with weighted round robin service in ATM local area networks," *Proc. Engineering of Complex Computer Systems*, pp. 96-103, Nov. 1995.

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

[10] B. Andersson, S. Baruah, and J. Jonsson, "Static-priority scheduling on multiprocessors," *Proc. 22nd IEEE Real-Time Systems Symposium*, pp. 193-202, Dec. 2001.

[11] B. Andersson and J. Jonsson, "Fixed-priority preemptive multiprocessor scheduling: to partition or not to partition," *Proc. 7th Int. Conf. Real-Time Computing Systems and Applications*, pp. 337-346, Dec. 2000.

[12] T. P. Baker, "Multiprocessor EDF and deadline monotonic schedulability analysis," *Proc. 24th IEEE Int. Real-Time Systems Symposium*, pp. 120-129, Dec. 2003.

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

[14] 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.

[15] J. Y. Le Boudec, "Application of network calculus to guaranteed service networks," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 1087-1096, May 1998.
[16] J. Y. Le Boudec and P. Thiran, Network Calculus, a Theory of Deterministic Queuing Systems for the Internet. New York: Springer-Verlag, 2001.
[17] E. Chan, E. Chen, J. Cao, and C. Lee, "Timing properties of the FDDI-M medium access protocol," *Computer Journal*, vol. 40, no. 1, pp. 43-49, Jun. 1997.
[18] 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.
[19] S. Chang, Performance Guarantees in Communication Networks. New York: Springer-Verlag, 2000.
[20] B. Chen, and W. Zhao, "Properties of the timed token protocol," Technical Report 92-038, Dept. of Computer Science, Texas A&M Univ., Oct. 1992.
[21] B. Chen, G. Agrawal, and W. Zhao, "Optimal synchronous capacity allocation for hard real-time communications with the timed token protocol", *Proc. IEEE Real-Time Systems Symposium* (RTSS'92), pp. 198-207, Dec. 1992.
[22] D. Chen, A. K. Mok, and T.-W. Kuo, "Utilization bound revisited," *IEEE Trans. Computers*, vol. 52, no. 3, pp. 351- 361, Mar. 2003.
[23] R. L. Cruz, "A calculus for network delay, part I: network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.
[24] R. L. Cruz, "A calculus for network delay, part II: network analysis," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 132–141, Jan. 1991.
[25] R. L. Cruz, "Quality of service guarantees in virtual circuit switched networks," *IEEE J. Selected Areas in Communications*, vol. 13, no. 6, pp. 1048–1056, Aug. 1995.
[26] R. L. Cruz, "SCED+: Efficient management of quality of service guarantees," *Proc. IEEE INFOCOM 1998*, pp. 625–642, Mar. 1998.
[27] *FDDI Token ring media access control (MAC).* ANSI Standard X3.139, 1987.
[28] F. Feng, A. Kumar, and W. Zhao, "Bounding application-to-application delays for multimedia traffic in FDDI-based communication systems," *Proc. Multimedia Computing and Networking*, pp. 174-185, Mar. 1996.
[29] S. Funk, J. Goossens, and S. Baruah, "On-line scheduling on uniform multiprocessors," *Proc. IEEE Real-Time Systems Symposium*, pp. 183-192, Dec. 2001.
[30] D. T. Green and D. T. Marlow, "SAFENET--a LAN for navy mission critical systems," *Proc. 14th IEEE Annual Conference on Local Area Computer Networks*, pp. 340-346, Oct. 1989.
[31] R. M. Grow, "A timed token protocol for local area networks," *Proc. Electro'82, Token Access Protocols*, paper 17/3, May 1982.
[32] M. Hamdaoui , and P. Ramanathan, "Selection of timed token protocol parameters to guarantee message deadlines," *IEEE/ACM Trans. on Networking*, vol. 3, no.3, pp. 340-351, Jun. 1995.
[33] C.-C. Han , K. G. Shin, "A polynomial-time optimal synchronous bandwidth allocation scheme for the timed-token MAC protocol," *Proc. 14th Annual Joint*

Conference of the IEEE Computer and Communication Societies, pp. 875-882, Apr. 1995.

[34] C.-C. Han, and H.-Y. Tyan, "A better polynomial-time schedulability test for real-time fixed-priority scheduling algorithm," *Proc. 18th IEEE Real-Time Systems Symposium*, pp. 36-45, Dec. 1997.

[35] H. Hansson, M. Sjodin, and K. Tindell, "Guaranteeing real-time traffic through an ATM network," *Proc. 30th Hawaii International Conference on System Sciences: Advanced Technology Track,* vol. 5, pp. 44-53, Jan. 1997.

[36] IEEE/ANSI Standard 802.4--1985 "Token passing bus access method and physical layer specifications," *IEEE*, New York, 1985.

[37] M. J. Johnson, "Reliability mechanisms of the FDDI high bandwidth token ring protocol," *Comput. Net. ISDN Syst*., vol. 2, no. 2, pp. 121-131, Feb. 1986.

[38] M. J. Johnson, "Proof that timing requirements of the FDDI token ring protocols are satisfied," *IEEE Trans. Communications*, vol. COM-35, no. 6, pp. 620-625, Jun. 1987.

[39] T.-W. Kuo and A. K. Mok, "Load Adjustment in Adaptive Real-Time Systems," *Proc. Real-Time Systems Symposium,* pp. 160-171, Dec. 1991.

[40] S. Lauzac, R. Melhem, and D. Mossé, "An improved rate-monotonic admission control and is applications," *IEEE Trans. on Computers*, vol. 52, no. 3, pp. 337 – 350, Mar. 2003.

[41] C.-G. Lee, L. Sha, and A. Peddi, "Enhanced utilization bounds for QoS management," *IEEE Trans. Computers,* vol. 53, no. 2, pp. 187-200, Feb. 2004.

[42] J. P. Lehoczky, "Fixed priority scheduling of periodic task sets with arbitrary deadlines," *Proc. 11th IEEE Real Time Systems Symposium*, pp. 201-209, Dec. 1990.

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

[44] J. Leung and J. Whitehead, "On the complexity of fixed-priority scheduling of periodic real-time tasks," *Performance Evaluation*, vol. 2, no. 4, pp. 235-250, Dec. 1982.

[45] C. Li, R. Bettati, and W. Zhao, "New delay analysis in high speed networks," *Proc. Int. Conf. Parallel Processing*, pp. 454–461, Sept. 1999.

[46] J. Liebeherr, A. Burchard,Y. Oh, and S. H. Son, "New strategies for assigning real-time tasks to multiprocessor systems," *IEEE Trans. Computers*, vol. 44, no. 12, pp. 1429–1442, Dec. 1995.

[47] J. Liebeherr, D. E. Wrege, and D. Ferrari, "Exact admission control for networks with a bounded delay service," *IEEE/ACM Trans. Networking*, vol. 4, no. 6, pp. 885–901, Jun. 1996.

[48] C. C. Lim, L. Yao, and W. Zhao, "A comparative study of three token ring protocols for real-time communications," *Proc. IEEE Conf. Distributed Computing Systems*, pp. 308-317, May 1991.

[49] 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.

[50] J. W. S. Liu, *Real-Time Systems.* Upper Saddle River, NJ: Prentice-Hall, 2000.

[51] L. Lundberg and H. Lennerstad, "Global multiprocessor scheduling of aperiodic tasks using time-independent priorities," *Proc. 9th IEEE Real-Time and Embedded Technology and Applications Symposium,* pp. 170-180, May 2003.

[52] N. Malcolm and W.Zhao, "Guaranteeing synchronous messages with arbitrary deadline constraints in an FDDI network," *Proc. IEEE Conf. Local Computer Networks*, pp. 186-195, Sept. 1993.

[53] N. Malcolm and W. Zhao, "The timed-token protocol for real-time communications," *Computer*, vol. 27, no. 1, pp. 35-41, Jan. 1994.

[54] N. Malcolm, and W. Zhao, "Hard real-time communication in multiple-access networks," *Real-Time Systems*, vol. 8, no. 1, pp. 35-77, Jan. 1995.

[55] N. Malcolm, S. Kamat, and W. Zhao, "Real-time communication in FDDI networks," *Real-Time Systems*, vol. 10, no. 1, pp. 75-107, Jan. 1996.

[56] A. K. Mok and D. Chen, "A general model for real-time tasks," Dept. Computer Sciences, The Univ. of Texas at Austin, Technical Report TR-96-24, Oct. 1996.

[57] 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.

[58] J. K.-Y. Ng, "The effect of different packet size in a time-token protocol network that supports real-time applications," *Proc. 2nd International Workshop on Real-Time Computing Systems and Applications*, pp. 34-41, Oct. 1995.

[59] J. K.-Y. Ng and V. C.-S. Lee, "Performance evaluation of transmission schemes for real-time traffic in a high-speed timed-token MAC network," *Journal of Systems and Software*, vol. 54, no. 1, pp. 41-60, Sept. 2000.

[60] J. K.-Y. Ng and J. Liu, "Performance of local area network protocols for hard real-time applications," *Proc. IEEE Conf. Distributed Computing Systems*, pp. 318-326, May 1991.

[61] J. K.-Y. Ng and G. Nong, "Multimedia applications on FDDI networks connected by an ATM switch," *Proc. IEEE Real-Time Technology and Applications Symposium*, pp. 60-65, May 1995.

[62] 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.

[63] J. L. Paige, "SAFENET--A navy approach to computer networking," *Proc. IEEE Conf. Local Computer. Net.*, pp. 268-273, Sept. 1990.

[64] 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.

[65] D.-W. Park, S. Natarajan, A. Kanevsky, and M. J. Kim, "A generalized utilization bound test for fixed-priority real-time scheduling," *Proc. 2nd International Workshop on Real-Time Computing Systems and Applications,* pp. 73-79, Oct. 1995.

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

[67] H. Sariowan, "A service curve approach to performance guarantees in integrated service networks," Ph.D. dissertation, Dept. Electrical and Computer Engineering, Univ. California, San Diego, 1996.
[68] H. Sariowan, R. L. Cruz, and G. C. Polyzos, "Scheduling for quality of service guarantees via service curves," *Proc. 1995 Int. Conf. Computer Communications and Networks*, pp. 512-520, Sept. 1995.
[69] K. G. Shin and Q. Zheng, "FDDI-M: a scheme to double FDDI's ability of supporting synchronous traffic," *IEEE Transactions on Parallel and Distributed Systems*, vol. 6, no. 11, pp. 1125-1131, Nov. 1995.
[70] J. K. Strosnider, J. Lehoczky, and L. Sha, "Advanced real-time scheduling using the IEEE 802.5 token ring," *Proc. IEEE Real-Time Systems Symposium*, pp. 42-52, Dec. 1988.
[71] A. Valenzano, P. Montuschi, and L. Ciminiera, "Some properties of timed token medium access protocols," *IEEE Trans. Software Eng.*, vol. 16, no. 8, pp. 858-869, Aug. 1990.
[72] S. Wang, D. Xuan, R. Bettati, and W. Zhao, "Providing absolute differentiated services for real-time application in static-priority scheduling networks," *Proc. IEEE INFOCOM*, pp. 669–678, Apr. 2001.
[73] 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,* pp. 529-540, Mar. 2005.
[74] 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*, pp. 251–262, Aug. 2000.
[75] 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, Dec. 1995.
[76] S. Zhang and A. Burns, "Timing properties of the timed token MAC protocol," *Proc. 6th International Conference on Computer Communications and Networks*, pp. 481-487, Sept. 1997.
[77] S. Zhang, A. Burns, and T.-H. Cheng, "Cycle-time properties of the timed token medium access control protocol," *IEEE Transactions on Computers*, vol. 51, no. 11, pp. 1362-1367, Nov. 2002.
[78] S. Zhang, A. Burns, J. Chen, and E. S. Lee, "Hard real-time communication with the timed token protocol: current state and challenging problems," *Real-Time System,* vol. 27, no. 3, pp. 271-295, Sept. 2004.
[79] 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.
[80] S. Zhang and E. S. Lee, "Efficient global allocation of synchronous bandwidths for hard real-time communication with the timed token MAC protocol," *Proc. 6th Int. Conf. Real-Time Computing Systems and Applications,* pp. 472-479, Dec. 1999.

[81] S. Zhang and E. S. Lee, "Determining the worst-case synchronous message response time in FDDI networks," *Computer Journal*, vol. 44, no. 1, pp. 31-41, Nov. 2001.

[82] S. Zhang, X. Liu, and E. S. Lee, "Effect of nodes reordering on the schedulability of real-time messages in timed token networks," *IEEE Communications Letters*, vol. 4, no. 12, pp. 420-422, Dec. 2000.

[83] 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.

APPENDIX A

LIST OF LEMMAS

Several supporting lemmas are needed in the proofs of the major theorems.

Lemma A-1. Given s-shaped functions $F_1,\ F_2,\ ...,F_h$,

$$\max_{0\le x\le(\ell+a)b}\left(x-\sum_{i=1}^{h}F_i(x)\right)\ge(\ell+1)\cdot\max_{0\le x\le ab}\left(x-\sum_{i=1}^{h}F_i(x)\right), \qquad \text{(A-1)}$$

where a and b are positive real numbers, a≤1, and ℓ is a non-negative integer.

Proof. First we claim that for any positive integer ℓ and s-shaped workload constraint function F

$$F(\ell x)\le \ell F(x). \qquad \text{(A-2)}$$

Let S be the segment length of F, and let $x=mS-\omega$, where $0\le\omega<S$, and m is an integer. Then by (II-11),

$$F(x)=F\left(mS\right). \qquad \text{(A-3)}$$

Since $\ell x=\ell(mS-\omega)\le \ell mS$, we have

$$F(\ell x)\le F(\ell mS). \qquad \text{(A-4)}$$

By Lemma 2-1,

$$F(\ell mS) \le \ell F(mS). \tag{A-5}$$

By substituting (A-4) and (A-3) into (A-5), we have (A-2) and thus the claim.

For convenience, we define

$$Z = \frac{\max_{0 \le x \le (\ell + a)b}\left(x - \sum_{i=1}^{h} F_i(x)\right)}{\ell + 1}. \tag{A-6}$$

Since $a < 1$, we can rewrite (A-6) as

$$Z = \max_{0 \le x \le (\ell + a)b}\left(\frac{x}{\ell + 1} - \frac{\sum_{i=1}^{h} F_i(x)}{\ell + 1}\right). \tag{A-7}$$

Let $x' = x/(\ell + 1)$. By (A-2),

$$F_i(x) = F_i\left((\ell + 1)x'\right) \le (\ell + 1)F_i(x'). \tag{A-8}$$

By substituting (A-8) into (A-7) and rearranging it, we get

$$Z \ge \max_{0 \le x' \le \frac{\ell + a}{\ell + 1}b}\left(x' - \sum_{i=1}^{h} F_i(x')\right). \tag{A-9}$$

Because $(\ell + a)/(\ell + 1) \ge a$, and because reducing the range the max operation of the right hand side of (A-9) will not increase its value, we have

$$Z \ge \max_{0 \le x' \le ab}\left(x' - \sum_{i=1}^{h} F_i\left(x'\right)\right). \tag{A-10}$$

Substituting (A-10) into (A-6), we have (A-2) established. □

Lemma A-2. Given positive real numbers $x_1,\ x_2,\ ...,\ x_h$, and a

$$\sum_{i=1}^{h-1} \frac{x_{i+1}}{x_i} + a\frac{x_1}{x_h} \geq ha^{\frac{1}{h}}. \tag{A-11}$$

Proof. Let $R\left(x_1,\ x_2,\ ...,x_h\right) = \sum_{i=1}^{h-1} \frac{x_{i+1}}{x_i} + a\frac{x_1}{x_h}$. Solving $\frac{\partial R}{\partial x_i} = 0$ for $i = 1,\ 2,\ ...,\ h$, we can verify that R will be minimized when $x_{i+1} / x_i = a^{\frac{1}{h}}$. □

Lemma A-3. Consider positive real numbers $x_1,\ x_2,...,\ x_h,\ y_1,\ y_2,\ ...,\ y_h$. If $x_i \geq x_{i+1} \geq 0$, for $i = 1,\ 2,\ ...,\ h-1$, and $\sum_{i=1}^{j} y_i \geq 0$, for $j = 1,\ 2,\ ...,\ h$, then

$$\sum_{i=1}^{h} \left(x_i y_i\right) \geq 0. \tag{A-12}$$

Proof. We will prove the lemma by induction on h. It is obvious that the lemma holds for $h = 1$. Assume that the lemma holds for $h = m$. We need to show that when $h = m+1$, the lemma also holds. First, we note

$$\sum_{i=1}^{m+1} \left(x_i y_i\right) = \sum_{i=1}^{m} \left(x_i y_i\right) + x_{m+1} y_{m+1}. \tag{A-13}$$

Since $\sum_{i=1}^{m+1} y_i \geq 0$, we have

$$y_{m+1} \geq -\sum_{i=1}^{m} y_i. \tag{A-14}$$

By substituting (A-14) into (A-13) and after some algebraic rearrangement, we get

$$\sum_{i=1}^{m+1}\left(x_i y_i\right) \geq \sum_{i=1}^{m}\left((x_i - x_{m+1})y_i\right). \tag{A-15}$$

Let $x_i' = x_i - x_{m+1}$, and since $x_i \geq x_{i+1} \geq 0$, we know that $x_i' \geq x_{i+1}' \geq 0$, $i = 1,\ 2,\ ...,\ m$. Then, by the induction hypothesis,

$$\sum_{i=1}^{m}\left((x_i - x_{m+1})y_i\right) = \sum_{i=1}^{m}\left(x_i' y_i\right) \geq 0. \tag{A-16}$$

By substituting (A-16) into (A-15), we establish (A-12) for $h = m+1$. Then the lemma is proven. □

Lemma A-4. Consider positive real numbers $x_1,\ x_2,\ ...,\ x_h,\ y_1,\ y_2,\ ...,\ y_h$, a, and b. If $0 \leq x_i \leq x_{i+1}$ for $i = 1,\ 2,\ ...,\ h-1$ and $x_i \leq a \leq (1+1/b)x_i$ for $i = 1,\ 2,\ ...,\ h$, then

$$\sum_{i=1}^{h}\frac{y_i}{a} - \sum_{i=1}^{h}\frac{y_i}{x_i} + \frac{1}{ab}\max_{j=1,\ 2,\ ...,\ n}\left(\sum_{i=1}^{j-1} y_i\right) \geq 0. \tag{A-17}$$

Proof. Let

$$Z = \sum_{i=1}^{h}\frac{y_i}{a} - \sum_{i=1}^{h}\frac{y_i}{x_i} + \frac{1}{ab}\max_{j=1,\ 2,\ ...,\ n}\left(\sum_{i=1}^{j-1} y_i\right). \tag{A-18}$$

Let ℓ be the value such that

$$\max_{i=1,\ 2,\ ...,\ n}\left(\sum_{i=1}^{j-1} y_i\right) = \sum_{i=1}^{\ell-1} y_i \,. \tag{A-19}$$

Then, ℓ must satisfy, for $m = 1,\ 2,\ ...,\ \ell$,

$$\sum_{i=m}^{\ell} y_i \geq 0 \tag{A-20}$$

and, for $m = \ell+1,\ \ell+2,\ ...,\ h$,

$$\sum_{i=\ell+1}^{m} y_i < 0 \,. \tag{A-21}$$

By substituting (A-19) into (A-18), we have

$$Z = \sum_{i=1}^{h} \frac{y_i}{a} - \sum_{i=1}^{h} \frac{y_i}{x_i} + \frac{1}{ab} \sum_{i=1}^{\ell-1} y_i . \tag{A-22}$$

Rewrite (A-22) as

$$Z = \sum_{i=1}^{\ell-1} \left(\left(\frac{b+1}{ab} - \frac{1}{x_i} \right) y_i \right) + \sum_{i=\ell}^{h} \left(\left(\frac{1}{x_i} - \frac{1}{a} \right) (-y_i) \right). \tag{A-23}$$

Let $x_i' = \frac{b+1}{ab} - \frac{1}{x_i}$. Since $x_i \leq a \leq \frac{b+1}{b} x_i$ and $x_i \leq x_{i+1}$, we have

$$x_{i+1}' \geq x_i' \geq 0 \,. \tag{A-24}$$

From (A-24), (A-20), and Lemma A-3, we have

$$\sum_{i=1}^{\ell-1}\left(\left(\frac{b+1}{ab}-\frac{1}{x_i}\right)y_i\right)=\sum_{i=1}^{\ell-1}\left(x_i\,'y_i\right)\geq 0. \tag{A-25}$$

Similarly, we can show

$$\sum_{i=\ell}^{h}\left(\frac{1}{x_i}-\frac{1}{a}\right)(-y_i)\geq 0. \tag{A-26}$$

By substituting (A-26) and (A-25) into (A-20), we get $Z \geq 0$, where Z is defined in (A-18). □

Lemma A-5. Consider s-shaped functions $F_1,\ F_2,\ \ldots,\ F_{h-1}$, and positive real numbers a and b. If $a \geq b$ and $a \geq S_j$, for $j = 1,\ 2,\ \ldots,\ h-1$, then

$$\sum_{j=1}^{h-1}\frac{F_j(\ell S_j)}{\ell S_j}+\frac{\max_{0\leq x\leq b}\left(x-\sum_{j=1}^{h-1}F_j(x)\right)}{a}\geq \min\left(\frac{b}{a},\ q\left(h\left((\frac{q+1}{q}\frac{b}{a})^{\frac{1}{h}}-1\right)+1-\frac{b}{a}\right)\right), \tag{A-27}$$

where $\ell = \max\left(1,\ \left\lfloor b/\max_{j=1,\,2,\,\ldots,\,h\text{-}1}\left(P_j\right)\right\rfloor\right)$, and

$$q = min_{j=1,\,2,\,\ldots,\,h-1}\left(\frac{F_j(\ell S_j)}{F_j((\ell+1)S_j)-F_j(\ell S_i)}\right).$$

Proof. By a careful observation of (A-27), we notice that (A-27) does not depend on the labeling order of F_j s. Thus, we can re-label the F_j such that, for $j = 1,\ 2,\ \ldots,\ h-2$,

$$S_j \leq S_{j+1}. \tag{A-28}$$

For the purpose of convenience, we define

$$Z(F_1,\ F_2,\ ...,F_{h-1},\ a,\ b)=\sum_{j=1}^{h-1}\frac{F_j(\ell S_j)}{\ell S_j}+\frac{\max_{0\le x\le b}\left(x-\sum_{j=1}^{h-1}F_j(x)\right)}{a}. \tag{A-29}$$

In the following proof, when the context is clear, we may use Z to denote $Z(F_1,\ F_2,\ ...,F_{h-1},\ a,\ b)$. Now we will prove the lemma in two cases, namely, $b\ge S_{h-1}$ and $b<S_{h-1}$.

Case 1: $b\ge P_{h-1}$**.** First, we define a new variable

$$S_h=b/\ell. \tag{A-30}$$

From the definition of ℓ, we know that for $j<h$

$$S_h\ge S_j. \tag{A-31}$$

A challenge in deriving a lower bound for Z is to remove the ceiling operations in the s-shaped functions F_i s. To do so, we will first impose a tighter restriction on the periods of F_i s, namely,

$$1\le\frac{S_h}{S_1}<\frac{\ell+1}{\ell}. \tag{A-32}$$

This restriction will be removed later. Now, let us define q_j, j = 1, 2, …, h,

$$q_j = \frac{F_j(\ell S_j)}{F_j((\ell+1)S_j) - F_j(\ell S_j)}. \tag{A-33}$$

By definitions of q_j and q,

$$q = \min_{j=1,\ 2\ ...,\ h-1}\left(q_j\right) \geq 1. \tag{A-34}$$

Now, if we let x in (A-29) take only the values of ℓS_j, $j = 1,\ 2,\ ...,\ h$, then the value of the right hand side of (A-29) will not increase. Thus,

$$Z \geq \sum_{i=1}^{h-1} \frac{F_i(\ell S_i)}{\ell S_i} + \frac{\max_{j=1,\ 2,\ ...,\ h}\left(\ell S_j - \sum_{i=1}^{h-1} F_i(\ell S_j)\right)}{a}. \tag{A-35}$$

By (A-28) and (A-32), we have, for $i \geq j$,

$$F_i(\ell S_j) \leq F_i(\ell S_i) \tag{A-36}$$

and, for $i < j$,

$$F_i(\ell S_j) \leq F_i\left((\ell+1)S_i\right). \tag{A-37}$$

Substituting (A-36) and (A-36) into (A-35), we get

$$Z \geq \sum_{i=1}^{h-1} \frac{F_i(\ell S_i)}{\ell S_i} + \frac{\max_{j=1,\ 2,\ ...,\ h}\left(\ell S_j - \sum_{i=1}^{j-1} F_i((\ell+1)S_i) - \sum_{i=j}^{h-1} F_i(\ell S_i)\right)}{a}. \tag{A-38}$$

Now, we will prove (A-38) for three sub-cases: $a \le (1+1/q)\ell S_1$, $(1+1/q)\ell S_1 < a \le (1+1/q)\ell S_{h-1}$, $a > (1+1/q)\ell S_{h-1}$.

Case 1.a: $a \le (1+1/q)\ell S_1$**.** Due to the facts that $a \ge b$ and $b = \ell S_h$,

$$a \ge \ell S_h . \tag{A-39}$$

We can rewrite (A-38) by moving $F_i(\ell S_i)$ out of the max operation and adding a term ℓS_1 into the max operation (with adding a term $\ell S_1 / a$ outside the max operation to balance). That is,

$$\begin{aligned} Z \ge & \sum_{i=1}^{h-1}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i(\ell S_i)}{a}\right) \\ & + \frac{\max\limits_{j=1,\,2,\,\ldots,\,h}\left(\sum_{i=1}^{j-1}\big(\ell(S_{i+1}-S_i)-(F_i((\ell+1)S_i)-F_i(\ell S_i))\big)\right)}{a} + \frac{\ell S_1}{a}. \end{aligned} \tag{A-40}$$

Let us define ε_i

$$\varepsilon_i = \ell(S_{i+1}-S_i)-(F_i((\ell+1)S_i)-F_i(\ell S_i)). \tag{A-41}$$

Substituting (A-41) into (A-40), we have

$$Z \ge \sum_{i=1}^{n-1}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i(\ell S_i)}{a}\right) + \frac{\ell S_1}{a} + \frac{\max_{j=1,\,2,\,\ldots,\,n}\left(\sum_{i=1}^{j-1}\varepsilon_i\right)}{a}. \tag{A-42}$$

By (A-39), (A-28), and $F_i(\ell S_i) \ge 0$, we have

$$\frac{F_i(\ell S_i)}{\ell P_i} - \frac{F_i(\ell S_i)}{a} \geq 0. \tag{A-43}$$

By (A-34), we know that $q / q_i \leq 1$. Thus, by (A-43), we have

$$\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i(\ell S_i)}{a} \right) \geq \left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i(\ell S_i)}{a} \right) \frac{q}{q_i}. \tag{A-44}$$

Now let us substitute (A-44) into (A-42) and rearrange it as follows:

$$Z \geq \sum_{i=1}^{h-1} \left(\left(\frac{1}{S_i} - \frac{\ell}{a} \right) \frac{F_i(\ell S_i)}{\ell} \frac{q}{q_i} \right) + \frac{\ell S_1}{a} + \frac{\max_{j=1,\ 2,\ \ldots,\ h} \left(\Sigma_{i=1}^{j-1} \varepsilon_i \right)}{a}. \tag{A-45}$$

By (A-33) and (A-41), we can rewrite $F_i(\ell S_i)$ as

$$F_i(\ell P_i) = q_i \left(F_i((\ell+1)S_i) - F_i(\ell S_i) \right) = q_i \left(\ell (S_{i+1} - S_i) - \varepsilon_i \right). \tag{A-46}$$

Substituting (A-46) into (A-45) and rearranging it, we have

$$Z \geq q \left(\sum_{i=1}^{n-1} \left((\frac{1}{S_i} - \frac{\ell}{a})(S_{i+1} - S_i) \right) + \frac{\ell S_1}{qa} \right) + \omega, \tag{A-47}$$

where

$$\omega = \sum_{i=1}^{h-1} \frac{q\varepsilon_i}{a} - \sum_{i=1}^{h-1} \frac{q\varepsilon_i}{\ell S_i} + \frac{\max_{j=1,\ 2,\ \ldots,\ h} \left(\sum_{i=1}^{j-1} \varepsilon_i \right)}{a}. \tag{A-48}$$

Now we will show $\omega \geq 0$ by using Lemma A-4. Let $x_i = \ell S_i$, $y_i = q\varepsilon_i$, $a' = a$, $b' = q$, and $h' = h$. As such, (A-48) can be rewritten as follows:

$$\omega = \sum_{i=1}^{h'-1} \frac{y_i}{a'} - \sum_{i=1}^{h'-1} \frac{y_i}{x_i} + \frac{\max_{j=1,\,2,\,\ldots,\,h'}\left(\sum_{i=1}^{j-1} y_i\right)}{a'b'}. \tag{A-49}$$

By (A-28) we have $x_i \le x_{i+1}$ and $x_i \le a'$. By $\ell S_n \le a \le (1+1/q)\ell S_1$ (the assumption of Case 1.a) and (A-28), we have $a' \le (1+1/b')x_i$. Then by Lemma A-4, we have

$$\omega \ge 0. \tag{A-50}$$

Substituting (A-50) into (A-47), we get

$$Z \ge q\left(\sum_{i=1}^{h-1}\left((\frac{1}{S_i} - \frac{\ell}{a})(S_{i+1} - S_i)\right) + \frac{\ell S_1}{qa}\right). \tag{A-51}$$

We rewrite (A-51) as follows:

$$Z \ge q\left(\sum_{i=1}^{h-1} \frac{S_{i+1}}{S_i} + \frac{\ell}{a}\frac{(q+1)S_h}{q}\frac{S_1}{S_h} - \frac{\ell S_h}{a} - h + 1\right). \tag{A-52}$$

From Lemma A-2, we have

$$\sum_{i=1}^{h-1} \frac{S_{i+1}}{S_i} + \frac{\ell}{a}\frac{(q+1)S_n}{q}\frac{S_1}{S_h} \ge h\left(\frac{(q+1)\ell S_h}{qa}\right)^{\frac{1}{h}}. \tag{A-53}$$

By substituting (A-53) into (A-52), we get

$$Z \ge q\left(h\left(\frac{(q+1)\ell S_h}{qa}\right)^{\frac{1}{h}} - \frac{\ell S_h}{a} - h + 1\right). \tag{A-54}$$

Per our definition of S_h in (A-30), we have $b = \ell S_h$. Thus, (A-54) becomes,

$$Z \geq q\left(h\left(\left(\frac{q+1}{q}\frac{b}{a}\right)^{\frac{1}{h}} - 1\right) + 1 - \frac{b}{a}\right). \tag{A-55}$$

(A-55) is equivalent to (A-38). Thus, we establish the lemma for Case 1.a.

Case 1.b: $(1+1/q)\ell S_1 < a \leq (1+1/q)\ell S_{h-1}$. Let us first rewrite (A-38) as

$$Z \geq max_{j=1,\,2,\,\ldots,\,h}\left(\frac{\ell S_j}{a} + \sum_{i=1}^{j-1}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right) + \sum_{i=j}^{h-1}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i(\ell S_i)}{a}\right)\right). \tag{A-56}$$

Define

$$\Pi \equiv \left\{F_i \mid \ell S_i \leq \frac{q_i}{q_i+1}a\right\}, \tag{A-57}$$

where q_i is defined in (A-33). Π is a set of F_i s whose periods are no less than $aq_i/\left(\ell(q_i+1)\right)$. We can rewrite (A-33) as

$$F_i(\ell S_i) = \frac{q_i}{q_i+1}F_i((\ell+1)S_i). \tag{A-58}$$

By dividing ℓS_i on both sides of (A-58), we get

$$\frac{F_i(\ell S_i)}{\ell S_i} = \frac{q_i}{q_i+1}\frac{F_i((\ell+1)S_i)}{\ell S_i}. \tag{A-59}$$

By (A-57), we have, for $F_i \in \Pi$,

$$\frac{q_i}{q_i+1} \geq \frac{\ell S_i}{a}. \tag{A-60}$$

By substituting (A-60) into (A-59) , we have, for $F_i \in \Pi$,

$$\frac{F_i(\ell S_i)}{\ell S_i} \geq \frac{F_i((\ell+1)S_i)}{a}. \tag{A-61}$$

Since

$$\sum_{i=1}^{j-1}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right) = \sum_{F_i \notin \Pi,\ 1\leq i<j}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right) + \sum_{F_i \in \Pi,\ 1\leq i<j}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right) \tag{A-62}$$

From (A-61), we have, for $F_i \in \Pi$,

$$\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a} \geq 0. \tag{A-63}$$

By substituting (A-63) to (A-62), we have

$$\sum_{i=1}^{j-1}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right) \geq \sum_{F_i \notin \Pi,\ 1\leq i<j}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right). \tag{A-64}$$

By (A-61) we have, for $F_i \in \Pi$,

$$\frac{F_i(\ell S_i)}{\ell S_i} \geq \frac{F_i((\ell+1)S_i)}{a} \geq \frac{F_i(\ell S_i)}{a}. \tag{A-65}$$

Similarly, we have

$$\sum_{i=1}^{j-1}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i(\ell S_i)}{a}\right) \geq \sum_{F_i \notin \Pi,\ 1\leq i<j}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i(\ell S_i)}{a}\right). \tag{A-66}$$

By substituting (A-64) and (A-66) into (A-56), we get

$$Z \geq \max_{j=1,\,2,\,\ldots,\,h}\left(\begin{array}{c}\dfrac{\ell S_j}{a} + \displaystyle\sum_{F_i \notin \Pi,\ 1\leq i<j}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right) \\ + \displaystyle\sum_{F_i \notin \Pi,\ 1\leq i<j}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i(\ell S_i)}{a}\right)\end{array}\right). \tag{A-67}$$

As reducing the range of the max operation in (A-67) will not increase its value, we have

$$Z \geq \max_{j,\, F_j \notin \Pi}\left(\begin{array}{c}\dfrac{\ell S_j}{a} + \displaystyle\sum_{F_i \notin \Pi,\ i<j}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right) \\ + \displaystyle\sum_{F_i \notin \Pi,\ i<j}\left(\frac{F_i(\ell S_i)}{\ell S_i} - \frac{F_i((\ell+1)S_i)}{a}\right)\end{array}\right). \tag{A-68}$$

We can rewrite (A-68) as follows:

$$Z \geq \sum_{F_i \notin \Pi}\frac{F_i(\ell S_i)}{\ell S_i} + \frac{\max_{j,\, F_i \notin \Pi}\left(\ell S_j - \displaystyle\sum_{F_i \notin \Pi,\ i<j} F_i((\ell+1)S_i) - \sum_{F_i \notin \Pi,\ i<j} F_i(\ell S_i)\right)}{a}. \tag{A-69}$$

Note that for any $F_i \notin \Pi$, $\ell P_i > \frac{q_i}{q_i+1} a \geq \frac{q}{q+1} a$. That is $a \leq (1+1/q)\ell S_i$. In comparison of (A-69) with (A-38), we see that (A-69) meets the requirements of Case 1.a (i.e. $a \leq (1+1/q)\ell S_i$). Furthermore, the number of summation reduces from h to h - m where $m = \|\Pi\|$ is the size of Π. Then, following the same argument we made for Case 1.a, we have,

$$Z \geq q\left((h-m)\left(\left(\frac{q+1}{q}\frac{b}{a} \right)^{\frac{1}{h-m}} - 1 \right) + 1 - \frac{b}{a} \right), \tag{A-70}$$

It can be verified that the right hand side of (A-70) is an increasing function of m. Then by substituting $m \geq 0$ into (A-70), we establish the lemma for Case 1.b.

Case 1.c: $a > (1+1/q)\ell S_{n-1}$. The proof of this sub-case is similar to the proof of Case 1.b. The difference is that in this sub-case,

$$m = \|\Pi\| = \text{h}, \tag{A-71}$$

where Π is defined in (A-57). Then (A-67) becomes

$$Z \geq \max_{j=1,\,2,\,\ldots,\,h}\left(\frac{\ell S_j}{a} \right) = \frac{\ell S_h}{a} = \frac{b}{a}. \tag{A-72}$$

Thus, we establish the lemma for Case 1.c.

We have proved the lemma for Case 1 with the constraint of $1 \le S_h / S_i \le 1 + 1/\ell$. Now, let us remove this constraint. Suppose $S_h / S > 1 + 1/\ell$, and F_i has period S_i and increments C_i^1, C_i^2, ..., C_i^L, C_i. Let $\ell_i = \lfloor S_h / S_i \rfloor$. Note that S_h is defined in (A-30). By (A-30), we have $\ell_i \ge \ell$. We construct F_i' with periods S_i' and increments $C_i^{'1}$, $C_i^{'2}$, ..., $C_i^{'L}$, $C_i^{'}$, where $S_i' = (\ell_i / \ell) S_i$, $C_i^{'j} = (\ell_i / \ell) C_i^j$, $j = 1, 2, ..., L$, and $C_i' = (\ell_i / \ell) C_i$. Furthermore, we define $S_h' = S_h$. Then, it can be verified that, for $i = 1, 2, ..., h\text{-}1$,

$$1 \le \frac{S_h'}{S_i'} < \frac{\ell + 1}{\ell} \tag{A-73}$$

and

$$\frac{F_i'(\ell S_i')}{\ell S_i'} = \frac{F_i(\ell S_i)}{\ell S_i}. \tag{A-74}$$

Based on Lemma 2-1, and by the fact that $\ell_i \ge \ell$, we know that if $x \in (\ell_i S_i,\ \ell S_h]$,

$$F_i(x) \le F_i((\ell_i + 1)S_i) \le \frac{\ell_i + 1}{\ell + 1} F_i((\ell + 1)S) \le \frac{\ell_i}{\ell} F_i((\ell + 1)S_i) = F'_i(x) \tag{A-75}$$

and if $x \in ((\ell - 1)S_h,\ \ell_i S_i]$,

$$F_i(x) \le F_i(\ell_i S_i) \le \frac{\ell_i}{\ell} F_i((\ell + 1)S_i) = F'_i(x) \tag{A-76}$$

Combining (A-76), (A-75) and (A-74) with (A-38), we have for all x, $(\ell - 1)S_h \le x \le \ell S_h$,

$$Z(F_1,\ F_2,\ ...,F_{h-1},\ a,\ b) \geq Z(F_1',\ F_2',\ ...,F_h',\ a,\ b). \tag{A-77}$$

By the results of Case 1.a, 1.b, 1.c, we have

$$Z(F_1',\ F_2',\ ...,F_{h-1}',\ a,\ b) \geq \min\left(\frac{b}{a},\ q\left(h\left((\frac{q+1}{q}\frac{b}{a})^{\frac{1}{h}}-1\right)+1-\frac{b}{a}\right)\right). \tag{A-78}$$

By substituting (A-78) into (A-29), we complete the proof for this case.

Case 2: $b < S_{h-1}$. By definition, we have

$$\ell = 1\,. \tag{A-79}$$

If $b < S_1$, then (A-38) can be verified by letting $x = b$. Now we will focus on the case $S_1 < b < S_{h-1}$. Let $m = \max_{i=1,\ 2,\ ...,\ h-1}\left(i \mid S_i \leq b\right)$. Recall that $a \geq S_i$. Consequently, for $i \geq m$ and $x \in [0,\ b]$, we have $b < S_i$ and

$$\frac{F_i(x)}{a} \leq \frac{F_i(b)}{a} \leq \frac{F_i(S_i)}{S_i}\ . \tag{A-80}$$

By substituting (A-80) into (A-29), we have

$$Z \geq \sum_{j=1}^{m}\frac{F_i(S_i)}{S_i} + \frac{\max_{0\leq x\leq b}\left(x - \sum_{i=1}^{m} F_i(x)\right)}{a}. \tag{A-81}$$

The right hand side of (A-81) satisfies the constraint of Case 1 for m s-shaped functions. Thus, we have

$$Z \geq \min\left(\frac{b}{a}, q\left(m\left((\frac{q+1}{q}\frac{b}{a})^{\frac{1}{m}} - 1\right) + 1 - \frac{b}{a}\right)\right) \tag{A-82}$$

Since the right hand side of the above inequality is a non-increasing function of m, we rewrite it into

$$Z \geq \min\left(\frac{b}{a}, q\left(n\left((\frac{q+1}{q}\frac{b}{a})^{\frac{1}{n}} - 1\right) + 1 - \frac{b}{a}\right)\right). \tag{A-83}$$

Lemma A-5 now is established. □

APPENDIX B

PROOF OF THEOREM 3-4

Theorem 3-4. Given a static priority scheduler, a schedulability bound with $\theta = 1/\lambda$ is given by

$$W^*\left(\frac{1}{\lambda}\right) = \begin{cases} 1 & h \le \frac{r}{r+1} \\ \frac{r}{h}\left(n\left((\frac{r+1}{r}h)^{\frac{1}{n}} - 1\right) + 1 - h\right) & \frac{r}{r+1} < h \le 1 \\ rn\left((\frac{r+1}{r})^{\frac{1}{n}} - 1\right) & h = 2, 3, \ldots \end{cases} \quad \text{(B-1)}$$

where k is the normalized deadline defined in (III-8), $r = k\eta$, η is the heterogeneity of tasks defined in (III-11), and h is

$$h = \frac{k}{\lambda}. \quad \text{(B-2)}$$

Proof. From Theorem 2-3, we have

$$W^*\left(\tfrac{1}{\lambda}\right) \ge \min_{\Gamma \in \Omega}\left(min_{i=1,\,2,\,\ldots,\,n}\left(min_{m=0,\,1,\,\ldots}\left(Z(i,\ m,\ \Gamma)\right)\right)\right), \quad \text{(B-3)}$$

where

$$Z(i,\ m,\ \Gamma) = \sum_{j=1}^{i-1} \frac{F_j\left(\frac{D_j}{\lambda}\right)}{\frac{D_j}{\lambda}} + \frac{max_{0 \le x \le mS_i + kS_i}\left(x - \sum_{j=1}^{i-1} F_j(x)\right)}{\frac{D_i}{\lambda} + m \cdot \min\left(\frac{D_i}{\lambda},\ S_i\right)}. \quad \text{(B-4)}$$

When the context of discussion in the following proof is clear, we will use Z to stand for $Z(i,\ m,\ \Gamma)$. In the rest of the proof, we will try to derive a lower bound of (B-4) in form of (B-1). We will consider for two cases: $k \le \lambda$ and $k = \Delta\lambda$, separately.

Case 1: $k \le \lambda$. Since $D_i / \lambda = kS_i / \lambda \le S_i$, we have $F_j\left(D_j / \lambda\right) = F_j(S_j)$ and we can rewrite (B-4) as

$$Z = \frac{\lambda}{k}\left(\sum_{j=1}^{i-1}\frac{F_j(S_j)}{S_j} + \frac{max_{0\le x\le(m+k)S_i}\left(x - \sum_{j=1}^{i-1}F_j(x)\right)}{(m+1)P_i}\right). \tag{B-5}$$

There will be three sub-cases here, depending on the value of k.

Case 1.a**:** $k \le 1$. By substituting the a, b, ℓ, and h in Lemma A-1 with k, S_i, m, and $i-1$, respectively, we have

$$\frac{max_{0\le x\le(m+k)P_i}\left(x - \sum_{j=1}^{i-1}F_j(x)\right)}{(m+1)S_i} \ge \frac{max_{0\le x\le kS_i}\left(x - \sum_{j=1}^{i-1}F_j(x)\right)}{S_i}. \tag{B-6}$$

Since $\lambda \ge 1$, we have $\lambda S_i \ge S_i$. Hence,

$$\frac{max_{0\le x\le(m+k)P_i}\left(x - \sum_{j=1}^{i-1}F_j(x)\right)}{(m+1)S_i} \ge \frac{max_{0\le x\le kS_i}\left(x - \sum_{j=1}^{i-1}F_j(x)\right)}{\lambda S_i}. \tag{B-7}$$

Substituting (B-7) into (B-5), we have

$$Z \geq \frac{\lambda}{k}\left(\sum_{j=1}^{i-1}\frac{F_j(S_j)}{S_j}+\frac{max_{0\leq x\leq kS_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{\lambda S_i}\right). \tag{B-8}$$

Now let us define

$$\ell_i = \max\left(1,\ \left\lfloor kS_i / \max_{j=1,\,2,\,\ldots,\,i-1}\left(S_j\right)\right\rfloor\right). \tag{B-9}$$

By Lemma 2-1, we have for $j = 1, 2, \ldots, n$,

$$F_j(S_j) \geq F_j(\ell_i S_j)/\ell_i\,. \tag{B-10}$$

Substituting (B-10) into (B-8), we get

$$Z \geq \frac{\lambda}{k}\left(\sum_{j=1}^{i-1}\frac{F_j(\ell_i S_j)}{\ell_i S_j}+\frac{max_{0\leq x\leq kS_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{\lambda S_i}\right). \tag{B-11}$$

By definition of priority inversion ratio, we know that

$$\lambda S_i \geq \lambda_i S_i\,. \tag{B-12}$$

Substituting (III-8) into the right hand side of (B-12), we have, for $j = 1, 2, \ldots, i$,

$$\lambda S_i \geq S_j\,. \tag{B-13}$$

We would like to use Lemma A-5 to find a lower bound of (B-11) by substituting a, b, h, and ℓ in Lemma A-5 with λS_i, kS_i, i, and ℓ_i, respectively. After doing that, we need to verify the conditions of Lemma A-5 hold. By (B-13), we know that $a = \lambda S_i \ge S_j$, $j = 1, 2, ..., h-1$. Since $k < 1$ (assumption of Case 1) and $\lambda \ge 1$, we have $a \ge b$. By (B-9), we get $\ell = \ell_i = \max\left(1, \lfloor b / \max_{i=1, 2, ..., h-1}(S_i) \rfloor\right)$. Then, by Lemma A-5, we get

$$Z \ge \frac{\lambda}{k} \min\left(\frac{k}{\lambda}, q\left(i\left((\frac{q+1}{q}\frac{k}{\lambda})^{\frac{1}{i}} - 1\right) + 1 - \frac{k}{\lambda}\right)\right), \tag{B-14}$$

where

$$q = \min_{j=1, 2, ..., i-1}\left(\frac{F_j(\ell S_j)}{F_j((\ell+1)S_j) - F_j(\ell S_j)}\right). \tag{B-15}$$

By definition of η, (III-11), (III-10), and $k < \lambda$, we have,

$$\eta = \min_{j=1, 2, ..., i-1}\left(\frac{F_j(S_j)}{F_j(2S_j) - F_j(S_j)}\right). \tag{B-16}$$

Since $\ell \ge 1$, we get

$$F_j(\ell S_j) \ge F_j(S_j) \tag{B-17}$$

By (II-12), we have

$$F_j((\ell+1)S_j) - F_j(\ell S_j) \le F_j(2S_j) - F_j(S_j) \tag{B-18}$$

Substituting (B-17) and (B-18) into (B-15), we get

$$q \geq \eta . \tag{B-19}$$

It can be verified that the right hand side of (B-14) is an increasing function of q, but a decreasing function of i. Hence, by substituting (B-19) and $i \leq n$ into (B-14) we get

$$Z \geq \min\left(1,\ \frac{\lambda}{k}\min\left(\frac{k}{\lambda},\ \eta\left(n\left((\frac{\eta+1}{\eta}\frac{k}{\lambda})^{\frac{1}{n}}-1\right)+1-\frac{k}{\lambda}\right)\right)\right). \tag{B-20}$$

Substituting (B-20) into (B-3), we have

$$W^{*}(1/\lambda) \geq \min\left(1,\ \frac{\lambda}{k}\min\left\{\frac{k}{\lambda}, \eta\left(n\left((\frac{\eta+1}{\eta}\frac{k}{\lambda})^{\frac{1}{n}}-1\right)+1-\frac{k}{\lambda}\right)\right\}\right). \tag{B-21}$$

This establishes (B-1)-(a) for Case 1.a.

Case 1.b**:** $1 < k \leq \lambda$, and there exists an h such that $(m+k)S_i < S_h$. By a similar argument made for (B-13), we have

$$\lambda S_i \geq \lambda_i S_i \geq S_h \geq (m+k)S_i \geq (m+1)S_i . \tag{B-22}$$

By substituting (B-22) into (B-5), we have

$$Z \geq \frac{\lambda}{k}\left(\sum_{j=1}^{i-1}\frac{F_j\left(S_j\right)}{S_j} + \frac{\max_{0\leq x\leq kS_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{\lambda S_i}\right). \tag{B-23}$$

Note that (B-23) has exactly the same form as (B-8). Then following the same argument made for deriving (B-20) from(B-8) , we have

$$Z \ge \frac{\lambda}{k}\min\left(\frac{k}{\lambda},\ \eta\left(i\left((\frac{\eta+1}{\eta}\frac{k}{\lambda})^{\frac{1}{i}}-1\right)+1-\frac{k}{\lambda}\right)\right). \qquad \text{(B-24)}$$

Thus, (B-1)-(a) is established for this sub-case as well.

Case 1.c: $1<k\le\lambda$, and $(m+k)S_i \ge S_j$, $j=1,\ 2,\ ...,\ i$. Since $k>1$, we have

$$\frac{\max_{0\le x\le(m+k)S_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{(m+1)S_i} \ge \frac{\max_{0\le x\le(m+k)S_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{(m+k)S_i}. \qquad \text{(B-25)}$$

Substituting (B-25) into (B-5), we have

$$Z \ge \frac{\lambda}{k}\left(\sum_{j=1}^{i-1}\frac{F_j(S_j)}{S_j}+\frac{\max_{0\le x\le(m+k)S_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{(m+k)S_i}\right). \qquad \text{(B-26)}$$

Now let us define

$$\ell_{i,\,m} = \min_{j=1,\,2,\,...,\,i-1}\left(\left\lfloor (m+k)S_i / S_j \right\rfloor\right). \qquad \text{(B-27)}$$

Note that $\ell_{i,\,m} \ge 1$. By Lemma 2-1, we have

$$F_i(S_i) \ge F_i(\ell_{i,\,m}S_i)/\ell_{i,\,m}\,. \qquad \text{(B-28)}$$

Substituting (B-28) into (B-23), we get

$$Z \geq \frac{\lambda}{k}\left(\sum_{j=1}^{i-1}\frac{F_j(\ell_{i,\,m}S_j)}{\ell_{i,\,m}S_j}+\frac{\max_{0\leq x\leq (m+k)S_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{(m+k)S_i}\right). \tag{B-29}$$

We would like to use Lemma A-5 to find a lower bound of (B-29) by substituting the a, b, h, and ℓ in Lemma A-5 with $(m+k)S_i$, $(m+k)S_i$, *i*, and $\ell_{i,\,m}$, respectively. Again, it is easy to verify that the conditions of Lemma A-5 hold after the substitution. Thus, by Lemma A-5, we get

$$Z \geq \frac{\lambda}{k}\min\left(1,\ qi\left((\frac{q+1}{q})^{\frac{1}{i}}-1\right)\right), \tag{B-30}$$

where

$$q = \min_{j=1,\,2,\,...,\,i-1}\left(\frac{F_j(\ell S_j)}{F_j((\ell+1)S_j)-F_j(\ell S_j)}\right). \tag{B-31}$$

Following the same argument we made when deriving (B-19) from (B-15), we have

$$q \geq \eta\,. \tag{B-32}$$

Note that the right hand side of (B-30) is an increasing function of q , but a decreasing function of *i*. Thus, by substituting (B-32) and $i \leq n$ into (B-30), we have

$$Z \geq \min\left(\frac{\lambda}{k},\ \frac{\lambda}{k}\eta n\left((\frac{\eta+1}{\eta})^{\frac{1}{n}}-1\right)\right) \geq \min\left(1,\ \frac{\lambda}{k}\eta n\left((\frac{\eta+1}{\eta})^{\frac{1}{n}}-1\right)\right). \tag{B-33}$$

By calculating the derivative of $n\left((\frac{\eta+1}{\eta}\frac{k}{\lambda})^{\frac{1}{n}}-1\right)+1-\frac{k}{\lambda}$ for k/λ, one can find that it is an increasing function of k/λ. Since $k \le \lambda$, we have

$$n\left((\frac{\eta+1}{\eta}\frac{k}{\lambda})^{\frac{1}{n}}-1\right)+1-\frac{k}{\lambda} \le n\left((\frac{\eta+1}{\eta})^{\frac{1}{n}}-1\right). \tag{B-34}$$

By substituting (B-34) into (B-33), we have

$$Z \ge \min\left(1,\ \frac{\lambda}{k}\eta\left(n\left((\frac{\eta+1}{\eta}\frac{k}{\lambda})^{\frac{1}{n}}-1\right)+1-\frac{k}{\lambda}\right)\right). \tag{B-35}$$

This establishes (B-1)-(a) for Case 1.c.

Case 2: $k = \Delta\lambda$, where Δ is a positive integer, $\Delta \ge 1$. We can rewrite (B-4) as

$$Z = \sum_{j=1}^{i-1}\frac{F_j\left(\Delta S_j\right)}{\Delta S_j} + \frac{\max_{0\le x\le (m+k)S_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{(m+\Delta)S_i}. \tag{B-36}$$

Since $\lambda \ge 1$ and $\Delta = k/\lambda$, we have $k \ge \Delta$. Substituting $k \ge \Delta$ into (B-36), we have

$$Z \ge \sum_{j=1}^{i-1}\frac{F_j\left(\Delta S_j\right)}{\Delta S_j} + \frac{\max_{0\le x\le (m+k)S_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{(m+k)S_i}. \tag{B-37}$$

Let us define

$$\ell_{i,\,m} = \left\lfloor (m+k)S_i / \max_{j=1,\,2,\,...,\,i-1}\left(S_j\right)\right\rfloor. \tag{B-38}$$

Knowing that $D_i = kS_i$, we know that $\lambda S_i \ge S_j$. Hence, for all $j = 1,\ 2,\ ...,\ i-1$,

$$(m+k)S_i \geq kS_i = \Delta\lambda S_i \geq \Delta S_j\,. \tag{B-39}$$

By substituting (B-39) into (B-38), we get

$$\ell_{i,\,m} \geq \Delta\,. \tag{B-40}$$

Note that Δ is a positive integer, per the assumption of this case. Then by Lemma 2-1, we have

$$\sum_{j=1}^{i-1} \frac{F_j\left(\Delta S_j\right)}{\Delta S_j} \geq \sum_{j=1}^{i-1} \frac{F_j\left(\ell_{i,\,m} S_j\right)}{\ell_{i,\,m} S_j}. \tag{B-41}$$

By substituting (B-41) into (B-37), we get

$$Z \geq \sum_{j=1}^{i-1} \frac{F_j\left(\ell_{i,\,m} S_j\right)}{\ell_{i,\,m} S_j} + \frac{\max_{0 \leq x \leq (m+k)S_i}\left(x - \sum_{j=1}^{i-1} F_j(x)\right)}{(m+k)S_i}. \tag{B-42}$$

Following the same argument we made when deriving (B-30) from (B-29), we get

$$Z \geq \min\left(1,\ qi\left(\left(\frac{q+1}{q}\right)^{\frac{1}{i}} - 1\right)\right) \geq qi\left(\left(\frac{q+1}{q}\right)^{\frac{1}{i}} - 1\right), \tag{B-43}$$

where

$$q = \min_{j=1,\,2,\,\ldots,\,i-1}\left(\frac{F_j(\ell_{i,\,m} S_j)}{F_j((\ell_{i,\,m}+1)S_j) - F_j(\ell_{i,\,m} S_j)}\right). \tag{B-44}$$

By definition of η, (III-11), and (III-10), we have

$$\eta = \frac{1}{\Delta} \min_{j=1,\ 2,\ \ldots,\ i-1} \left(\frac{F_j(\Delta S_j)}{F_j((\Delta+1)S_j) - F_j(\Delta S_j)} \right). \tag{B-45}$$

Following the same argument we made when deriving (B-19) from (B-15), we have

$$q \geq \Delta \eta. \tag{B-46}$$

It can be verified that the right hand side of (B-43) is an increasing function of q, but a decreasing function of i. Thus, by substituting (B-47) and $i \leq n$ into (B-43), we have

$$Z \geq n\Delta\eta \left(\left(\frac{\Delta\eta + 1}{\Delta\eta} \right)^{\frac{1}{n}} - 1 \right). \tag{B-47}$$

By substituting $\Delta = k / \lambda$ into (B-47), we establish (B-1)-(b). □

APPENDIX C

PROOF OF COROLLARY 3-4

Corollary 3-4. Given a non-preemptive static priority scheduler and a task set Γ with s-shaped workload constraint functions, Γ is schedulable if

$$W^*\left(\frac{1}{\lambda^*}\right)=\begin{cases}\min\left(1,\ \frac{\lambda^*}{k}\eta\left(n'\left((\frac{\eta+1}{\eta}\frac{k}{\lambda^*})^{\frac{1}{n'}}-1\right)+1-\frac{k}{\lambda^*}\right)\right) & k\le\lambda^*;\\ \frac{k}{\lambda^*}\eta n'\left((\frac{k\eta+\lambda^*}{k\eta})^{\frac{1}{n'}}-1\right) & k=\Delta\lambda^*,\text{ and }\Delta\text{ is a positive integer.}\end{cases} \quad \text{(C-1)}$$

where $\lambda^* = \max_{i=1,\,2,\,...,\,n}\{D_i\}/\min_{i=1,\,2,\,...,\,n}\{D_i\}$ and n' is the number tasks with non-dividable segment lengths.

Proof. From Theorem 2-3, we have

$$W^*\left(\tfrac{1}{\lambda^*}\right)\ge \min_{\Gamma\in\Omega}\left(min_{i=1,\,2,\,...,\,n}\left(min_{m=0,\,1,\,...}\left(Z(i,\ m,\ \Gamma)\right)\right)\right), \quad \text{(C-2)}$$

where

$$Z(i,\ m,\ \Gamma)=\sum_{j=1}^{i-1}\frac{F_j\left(\frac{D_j}{\lambda}\right)}{\frac{D_j}{\lambda^*}}+\frac{max_{0\le x\le mS_i+kS_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{\frac{D_i}{\lambda^*}+m\cdot\min\left(\frac{D_i}{\lambda^*},\ S_i\right)}. \quad \text{(C-3)}$$

Then by following the same argument we proving Corollary 3-4, we can merge the tasks with divisible segment lengths and reduce the task number from n to n'. Then, on the merged task set, we can follow the same argument in proving Theorem 3-4 to reach (C-1). □

APPENDIX D

DERIVATION OF LEOHCZKY'S BOUND

In CHAPTER III, during parametric fitting of several existing bounds, we pointed out that one of the bounds obtained from Theorem 3-4 is not exactly the same as the one derived by Leohczky in [34] (n v.s. $n-1$). This difference is due to the fact that Theorem 3 is derived for general static priority schedulers without using the case specific information. Here, we show how to derive the exact Leohczky's bound by using Theorem 3-4. Recall that in [34], the real-time system has a rate monotonic scheduler, n periodic tasks (S_i = P_i), and D_i = k P_i, for i = 1, 2, …, n. With this model we can establish the following Theorem.

Theorem D-6. Given a rate monotonic scheduler with set of periodic tasks, a schedulability bound with $\theta = 1/k$ is given by

$$W^*(1/k) \geq k(n-1)\left(\frac{k+1}{k}\right)^{\frac{1}{n-1}}. \qquad \text{(D-1)}$$

Proof. From Theorem 2-3,

$$W^*(1/k) \geq \min_{\Gamma\in\Omega}\left(\min_{i=1,\,2,\,\ldots,\,n}\left(\min_{m=0,\,1,\,\ldots}\left(Z(i,\ m,\ \Gamma)\right)\right)\right), \qquad \text{(D-2)}$$

where

$$Z(i,\ m,\ \Gamma) = \sum_{j=1}^{i-1} \frac{F_j\left(P_j\right)}{P_j} + \frac{\max_{0 \le x \le (m+k)P_i}\left(x - \sum_{j=1}^{i-1} F_j(x)\right)}{(m+1)P_i}. \tag{D-3}$$

When the context of discussion is clear, we will use Z to stand for $Z(i,\ m,\ \Gamma)$ in the following proof. Let us define ℓ_j, $j = 1, 2, \ldots, i-1$, as

$$\ell_j = \left\lfloor (m+k)P_i \big/ P_j \right\rfloor. \tag{D-4}$$

Then we have

$$(m+k)P_i \le (\ell_{i-1} + 1)P_{i-1}. \tag{D-5}$$

(D-5) can be rewritten as

$$\ell_{i-1}P_{i-1} \ge (m+k)P_i - P_{i-1}. \tag{D-6}$$

Recall that for a rate monotonic scheduler, the task priorities are assigned in the descending order of periods. That is,

$$P_i \ge P_{i-1} \ge \ldots \ge P_1. \tag{D-7}$$

Hence,

$$\ell_{i-1}P_{i-1} \ge (m+k-1)P_i. \tag{D-8}$$

By substituting $k \ge 2$ into (D-8), we have

$$\ell_{i-1}P_{i-1} \geq (m+1)P_i. \tag{D-9}$$

Substituting (D-9) into (D-3), we get

$$Z \geq \sum_{j=1}^{i-1}\frac{F_j\left(P_j\right)}{P_j}+\frac{\max_{0\leq x\leq(m+k)P_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{\ell_{i-1}P_{i-1}}. \tag{D-10}$$

By (D-4),

$$\ell_{i-1}P_{i-1} \leq (m+k)P_i. \tag{D-11}$$

Reducing the range of the max operation in (D-10) does not increase its value. As a result, we have

$$Z \geq \sum_{j=1}^{i-1}\frac{F_j\left(P_j\right)}{P_j}+\frac{\max_{0\leq x\leq\ell_{i-1}P_{i-1}}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{\ell P_{i-1}}. \tag{D-12}$$

Since $x \leq \ell_{i-1}P_{i-1}$,

$$F_{i-1}(x) \leq F_{i-1}(\ell_{i-1}P_{i-1}). \tag{D-13}$$

Substituting (D-13) into (D-12), we have

$$Z \geq \sum_{j=1}^{i-1}\frac{F_j\left(P_j\right)}{P_j}+\frac{\max_{0\leq x\leq\ell_{i-1}P_{i-1}}\left(x-\sum_{j=1}^{i-2}F_j(x)-F_{i-1}(\ell_{i-1}P_{i-1})\right)}{\ell_{i-1}P_{i-1}}. \tag{D-14}$$

Rewrite (D-14) into

$$Z \geq \sum_{j=1}^{i-2} \frac{F_j\left(P_j\right)}{P_j} + \frac{\max_{0 \leq x \leq \ell_{i-1} P_{i-1}} \left(x - \sum_{j=1}^{i-2} F_j(x)\right)}{\ell_{i-1} P_{i-1}} + \left(\frac{F_{i-1}(P_{i-1})}{P_{i-1}} - \frac{F_{i-1}(\ell_{i-1} P_{i-1})}{\ell_{i-1} P_{i-1}}\right). \quad \text{(D-15)}$$

Since the tasks are periodic, $F_{i-1}\left(\ell_{i-1} P_{i-1}\right) = \ell_{i-1} F_{i-1}\left(P_{i-1}\right)$. Hence,

$$Z \geq \sum_{j=1}^{i-2} \frac{F_j\left(P_j\right)}{P_j} + \frac{\max_{0 \leq x \leq \ell_{i-1} P_{i-1}} \left(x - \sum_{j=1}^{i-2} F_j(x)\right)}{\ell_{i-1} P_{i-1}}. \quad \text{(D-16)}$$

Then by Lemma A-5, we have

$$Z \geq \ell\left((i-1)\left(\left(\frac{\ell+1}{\ell}\right)^{\frac{1}{i-1}} - 1\right)\right), \quad \text{(D-17)}$$

where

$$\ell = \left\lfloor \ell_{i-1} P_{i-1} / \max_{j=1,\,2,\,\ldots,\,i\,-\,2}\left(P_j\right) \right\rfloor. \quad \text{(D-18)}$$

By (D-4) and (D-7), $\ell \geq k$. Since the right hand side of (D-17) is an increasing function of ℓ, but a decreasing function of i. By substituting $\ell \geq k$ and $i \leq n$ into (D-17), we have

$$Z \geq k(n-1)\left(\frac{k+1}{k}\right)^{\frac{1}{n-1}}. \quad \text{(D-19)}$$

Then, the theorem follows. □

APPENDIX E

PROOF OF COROLLARY 3-5

Corollary 3-5. Given a static priority scheduler and a task set Γ with s-shaped workload constraint functions, Γ is schedulable if

$$W^*\left(\frac{1}{\lambda}\right)=\begin{cases}\min\left(1,\ \frac{\lambda}{k}\eta\left(n'\left((\frac{\eta+1}{\eta}\frac{k}{\lambda})^{\frac{1}{n'}}-1\right)+1-\frac{k}{\lambda}\right)\right) & k\le\lambda;\\ \frac{k}{\lambda}\eta n'\left((\frac{k\eta+\lambda}{k\eta})^{\frac{1}{n'}}-1\right) & k=\Delta\lambda,\text{ and }\Delta\text{ is a positive integer.}\end{cases} \tag{E-1}$$

where n' is the number tasks with non-dividable segment lengths.

Proof. From Theorem 2-3, we have

$$W^*\left(\tfrac{1}{\lambda}\right)\ge\min_{\Gamma\in\Omega}\left(min_{i=1,\ 2,\ ...,\ n}\left(min_{m=0,\ 1,\ ...}\left(Z(i,\ m,\ \Gamma)\right)\right)\right), \tag{E-2}$$

where

$$Z(i,\ m,\ \Gamma)=\sum_{j=1}^{i-1}\frac{F_j\left(\frac{k}{\lambda}S_j\right)}{\frac{k}{\lambda}S_j}+\frac{max_{0\le x\le mS_i+kS_i}\left(x-\sum_{j=1}^{i-1}F_j(x)\right)}{\frac{k}{\lambda}S_j+m\cdot\min\left(\frac{k}{\lambda}S_j,\ S_i\right)}. \tag{E-3}$$

When the context of discussion in the following proof is clear, we will use Z to denote $Z(i,\ m,\ \Gamma)$.

Let construct a new task set Γ', by merging T_i, and T_j, where $S_i = mS_i$, m is an integer, into T_i' with $S_i' = S_i$, $F_i'(\ell S_i') = F_i(\ell S_i) + F_j(m\ell S_j)$, $\ell = 1, 2, \ldots$. It can be verified that

$$F_i(\tfrac{k}{\lambda} S_i) + mF_j\left(\tfrac{k}{\lambda} S_i\right) \geq F_i(\tfrac{k}{\lambda} S_i) + F_j\left(\tfrac{k}{\lambda} mS_j\right) = F_j'\left(\tfrac{k}{\lambda} S'_j\right). \tag{E-4}$$

and

$$F_i(x) + F_j(x) \geq F'_i(x). \tag{E-5}$$

So we have

$$Z(i,\ m,\ \Gamma) \geq Z(i,\ m,\ \Gamma'). \tag{E-6}$$

By continuing this process, we can find a task set Γ^* with n' tasks with n' periods that are non-dividable each other, yet

$$Z(i,\ m,\ \Gamma) \geq Z(i,\ m,\ \Gamma^*). \tag{E-7}$$

Then follow the same procedure in proving the Theorem 3-4, we have the theorem. □

VITA

Jianjia Wu was born in Inner Mongolia, the People's Republic of China. He received his Bachelor of Science degree from Tianjin University, Tianjin, China, in July 1997, and Master of Science degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in March 2000, both in computer science. His research interests include Real-Time Systems, Grid Computing, and Network Security. He can be reached care of Dr. Wei Zhao, Department of Computer Science, Texas A&M University, College Station, TX, 77845. His email address is jianjiawu@tamu.edu.