

A Unified and Parameterized Schedulability Bound of Static Priority Schedulers for Real-Time Systems¹

Jianjia Wu, Jyh-Charn Liu, *Member, IEEE*, and Wei Zhao, *Fellow, IEEE*

Abstract—In this paper, we analyze static priority schedulers for real-time tasks and derive a parameterized schedulability bound for arbitrary static priority schedulers using the general schedulability bound analysis framework proposed in [1]. The schedulability bound uses several parameters to represent important system behaviors: (1) Number of tasks; (2) Normalized deadline that measures the tightness of task deadlines; (3) Task set workload heterogeneity, which gauges the fitness of the workload descriptions; and (4) Deadline inversion ratio, which measures the degree of priority inversions. Schedulability bounds for specific system configurations can be easily obtained by simply plugging in proper parameters. We show that existing utilization bounds are points and curves on our new multi-dimension bound surface.

Index Terms—**Workload rate, utilization, schedulability test, static priority scheduler, network calculus**

¹ A shorter version of this work was presented in IEEE real-time and embedded technology and application symposium 2005, San Francisco, California [2]. Wu and Liu are with the Texas A&M University, College Station, TX 77843 USA. Wei Zhao is with Rensselaer Polytechnic Institute, Troy, NY 12180. Send email correspondence to zhaow3@rpi.edu.

I. INTRODUCTION

Meeting stringent timing requirements is critical to real-time applications. The utilization bound based *schedulability test* is a common approach to determine whether or not tasks can be admitted without violating deadline requirements. In [1], we introduced a general utilization bound analysis framework based on network calculus techniques and derived a general schedulability bound for arbitrary real-time systems. In this paper, we use this general framework and the bound result to analyze, in detail, the schedulability bound of static priority schedulers. We focus on static priority schedulers because they are widely used in real-time systems and many schedulability bounds have been derived [3-8]. These bound results serve as good comparison and evaluation bases for our new parameterized bound results.

We first introduce a special s-shaped workload constraint function, which is a generalization of the classical periodic task model and allows flexible characterization of the task workloads. The service constraint function is then derived for the rate monotonic scheduler (RMS). These functions are plugged into the general schedulability bound derived in [1] to obtain a closed-form schedulability bound for the rate monotonic scheduler with s-shaped workload constraint functions. The bound is parameterized for various task workload characteristics. We show that our bound generalizes the utilization bounds of the following systems which have been analyzed in previous studies. That is, when the parameters are properly set, our bound reduces to those obtained in these previous studies.

- The system has periodic tasks whose deadlines are equal to periods. Tasks are scheduled by a rate monotonic scheduler [5].
- The system has periodic tasks whose deadlines are less than periods. Tasks are scheduled

1 by a rate monotonic scheduler [4], [8].

2 • The system has periodic tasks whose deadlines are multiples of periods. Tasks are
3 scheduled by a rate monotonic scheduler [3].

4 • The system has multi-frame tasks whose deadlines are equal to periods. Tasks are
5 scheduled by a rate monotonic scheduler [7].

6 To our knowledge, no other research has explored as wide a range of systems as ours.

7 We then generalize the parameterized bound to static priority scheduler with arbitrary priority
8 assignment. This general schedulability bound is further improved based on the concept of
9 harmonic chain tasks [9-10]. Moreover, we propose an r-shaped workload constraint function,
10 which is a generalization of the leaky bucket traffic stream regulator widely used in a network
11 environment. A closed-formed schedulability bound is also derived for arbitrary static priority
12 schedulers with r-shaped workload constraint functions.

13 The rest of the paper is organized as follows. Section II introduces an s-shaped workload
14 constraint function and derives the service constraint functions for the rate monotonic scheduler.
15 A closed-formed schedulability bound for the rate monotonic scheduler is obtained in Section
16 III. Performance evaluation and comparison of the derived bound are provided in Section IV. In
17 Section V, the schedulability bound is further extended to other static priority systems, e.g. non-
18 preemptive systems, system with harmonic chain tasks [9-10], and system with r-shaped tasks.
19 The discussion and conclusion are provided in Section VI.

20 **II. Task and Scheduler Model**

21 **A. Terminologies**

1 In this paper, we will use the same terminologies as defined in [1]. Specifically, we will use
 2 $\Gamma = \{T_1, T_2, \dots, T_n\}$ to denote a task set, where T_i is the i^{th} task and D_i the relative deadline of T_i .
 3 We use $f(t)$ to denote the workload function for T ,

$$4 \quad f(t) = \text{the summation of the sizes of all the jobs from task } T \text{ in } [0, t], \quad (\text{II-1})$$

5 and $g(t)$ the service function for T ,

$$6 \quad g(t) = \text{the total execution time rendered to the jobs from task } T \text{ in } [0, t]. \quad (\text{II-2})$$

7 Let $F(I)$ denote the *workload constraint function* for T . $F(I)$ satisfies that $\forall I, 0 \leq I \leq t$

$$8 \quad f(t) - f(t - I) \leq F(I). \quad (\text{II-3})$$

9 We use $G(I)$ to denote the *service constraint function* for T . $G(I)$ satisfies that $\forall t \geq 0$, there
 10 exists $I \leq t$ such that

$$11 \quad g(t) \geq f(t - I) + G(I). \quad (\text{II-4})$$

12 Let $W(\theta, \Gamma)$ denote the *scaled workload rate* for task set Γ . $W(\theta, \Gamma)$ is calculated as

$$13 \quad W(\theta, \Gamma) = \sum_{i=1}^n \frac{F_i(\theta D_i)}{\theta D_i}, \quad (\text{II-5})$$

1 where θ , $\theta > 0$, is a scaling parameter.

2 For detailed discussion of these terminologies, see [1].

3 **B. S-shaped Workload Constraint Functions**

4 Knowing that workload constraint functions can exist in many different forms, we start with a
 5 special type of workload constraint function, namely the *staircase-shaped (s-shaped) workload*
 6 *constraint function*.

7 As its name suggests, an s-shaped workload constraint function consists of segmented pieces,
 8 and resembles a staircase. The values of an s-shaped workload constraint function increase only
 9 at border points of segments. We assume that the segment length S is fixed, and the increments
 10 may not be identical for the first L segments where L is a parameter in the function.

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

$$12 \quad F(I) = \begin{cases} \sum_{j=1}^a C^j & a \leq L \\ \sum_{j=1}^L C^j + (a-L)C & a > L, \end{cases} \quad (\text{II.6})$$

13 where $a = \lceil I/S \rceil$, C^j is the increment at the beginning of the j^{th} segment, and C is the
 14 constant increment after the L^{th} segment. Figure 1 shows an example of the s-shaped workload
 15 constraint function. When $L = \infty$, an s-shaped constraint function reduces to the general real-
 16 time task model proposed in [6]. In the following discussions, we call a task with an s-shaped
 17 workload constraint function an s-shaped task.

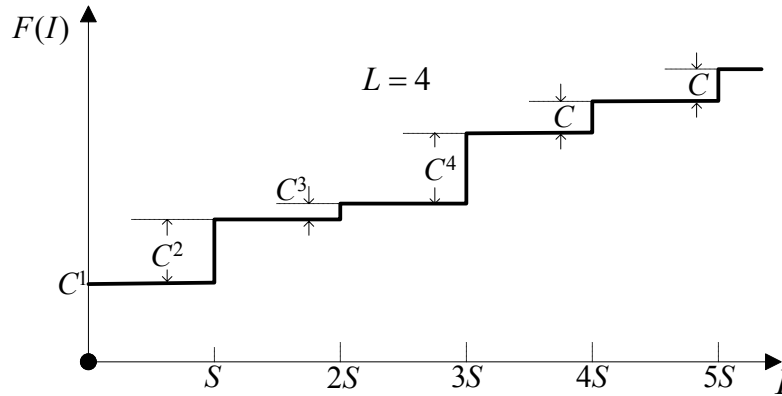


Fig. 1: An Example S-shaped Workload Constraint Function

1

2 We say that an s-shaped workload constraint function $F(I)$ is *smooth* if the following holds:

3

$$C^1 \geq C^2 \geq \dots \geq C^L \geq C. \quad (\text{II.7})$$

4 That is, if an s-shaped workload constraint function is smooth, then its increments over time are
 5 non-increasing. The smoothness property can simplify the schedulability bound analysis.

6 Any non-smooth s-shaped workload constraint function can be converted into a smooth one by
 7 the following procedure:

8 Step 1. Locate the first C^j in the function such that $C^j < C^{j+1}$;

9 Step 2. Replace both C^j and C^{j+1} with $(C^j + C^{j+1})/2$;

10 Step 3. Repeat Steps 1 and 2 until no such C^j exists;

11 Step 4. Replace the C^j s that are less than C with C .

12 It can be easily verified that the above process always produces a smooth s-shaped workload
 13 constraint function. For example, consider the non-smooth s-shaped constraint function in Figure

14 1 with segment length $S = 8$, $C^1 = 5$, $C^2 = 5$, $C^3 = 1$, $C^4 = 5$, and $C = 2$. This function can be
 15 transformed into a smooth one with $S = 8$, $C^1 = 5$, $C^2 = 5$, $C^3 = 3$, $C^4 = 3$, and $C = 2$. In the rest

1 of this paper, unless otherwise specified, we only consider smooth s-shaped workload constraint
 2 functions.

3 **C. Service Constraint Function of the Rate Monotonic Scheduler**

4 In a static priority system, each task is assigned a priority number. All the jobs from the same
 5 task have the same priority value and are executed in a first-come-first-served order. A job can
 6 execute only if no job with higher priority is waiting for the processor.

7 Without loss of generality, in the following discussions, we assume the tasks in a static priority
 8 system are labeled in ascending order of their priorities, i.e. T_1 has the highest priority.

9 **Theorem 1:** For a static priority scheduling system, a service constraint function for task T_i is

$$10 \quad G_i(I) = \sup_{0 \leq x \leq I} \left(x - \sum_{j=1}^{i-1} F_j(x) \right). \quad (II.8)$$

11 **Proof:** See [11].

12 Note that operation "sup" guarantees that $G_i(I)$ is non-negative and non-decreasing.

13 For s-shaped task, we say a static priority scheduler is rate monotonic if the task priorities are
 14 assigned in descending order of the segment lengths. That is, task T_i has a higher priority than
 15 T_j if $S_i < S_j$. For a rate monotonic scheduler, we have the following result on its service
 16 constraint functions.

17 **Corollary 1:** For a rate monotonic scheduler with s-shaped tasks, a service constraint function
 18 for task T_i is

$$G_i(I) = \sup_{0 \leq x \leq I} \left(x - \sum_{S_j \leq S_i} F_j(x) \right). \quad (\text{II.9})$$

Proof: Since the tasks are labeled in descending order of their priorities, i.e. T_1 has the highest priority, and a rate monotonic scheduler assigns higher priority to tasks with shorter segment lengths, we know that $S_j \leq S_i$ if and only if $j \leq i$. Together with Theorem 1, we immediately have (II.9).

III. Parameterized Schedulability Bound for the Rate Monotonic Scheduler

In this section, we derive a closed-formed parameterized schedulability bound for the rate monotonic schedulers with s-shaped tasks. First, we will introduce several key system parameters that measure the different system configurations.

A. Key System Parameters

For an s-shaped task T , we define its *normalized deadline*, k , as follows:

$$k = D / S, \quad (\text{III.1})$$

where D is the relative deadline of T and S is the segment parameter in the workload constraint function defined in (II.6).

k can be viewed as the deadline using S as the measurement unit, and it characterizes the tightness of the deadline requirement. The smaller the k , the more difficult it is to schedule the task. Following conventional literature [2-11], we assume that all the tasks have the same value of k . That is, for $i = 1, 2, \dots, n$

$$1 \quad k_i = k. \quad (III.2)$$

2 We know that the workload rate $W(\theta, \Gamma)$ is an upper bound of the job releasing rate averaged
 3 in a window of finite length θD_i . Since an s-shaped function allows bursty job release in short
 4 time windows, as long as the job release rate reduces in longer time windows, using the
 5 workload rate measured in a window of length θD_i for schedulability bound analysis may over-
 6 estimate the resource demand, and thus result in under-estimated schedulability bounds. To
 7 overcome this problem, in our schedulability bound analysis, we take into account the variance
 8 of the workload rate measured in different windows. We proposed here a task parameter,
 9 *workload heterogeneity* $V_i(\hbar)$ for this purpose, which is defined as:

$$10 \quad V_i(\hbar) = \frac{F_i(\hbar S_i) / \hbar}{F_i((\hbar+1)S_i) - F_i(\hbar S_i)}, \quad (III.3)$$

11 where \hbar is a positive integer. Intuitively, $V_i(\hbar)$ is the ratio between the workload rate measured
 12 in $[0, \hbar S_i]$, and the one measured in $[\hbar S_i, (\hbar+1)S_i]$ as shown in Figure 2. Clearly, for periodic
 13 task, $V_i(\hbar) = 1$.

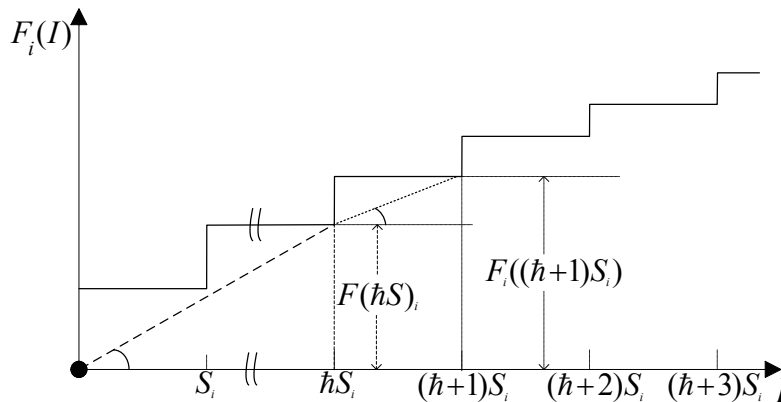


Fig. 2: Workload Heterogeneity.

14

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

$$2 \quad V(\mathbf{h}, \Gamma) = \min(V_1(\mathbf{h}), V_2(\mathbf{h}), \dots, V_n(\mathbf{h})). \quad (\text{III.4})$$

3 We note that a similar concept of workload heterogeneity was first introduced in [7].

4 **B. Parameterized Schedulability Bound for the Rate Monotonic Scheduler**

5 From Theorem 2 in [1], we know that a lower bound of schedulability bound for a scheduler
6 with respect to a task set space Ω is given by

$$7 \quad W^*(\theta) = \inf_{\Gamma \in \Omega} \left(\sum_{j=1}^n \frac{F_j(\theta D_j)}{\theta D_j} + \min_{i=1, 2, \dots, n} \left((\alpha_i - 1) \frac{F_i(\theta D_i)}{\theta D_i} \right) \right), \quad (\text{III.5})$$

8 where $\alpha_i = \inf_{I > 0} (G_i(I + D) / F_i(I))$. Note that (III.5) holds for any type of workload constraint
9 function. For s-shaped workload constraint functions, we have the following corollary.

10

11 **Theorem 2:** Given a scheduler and a collection of task set with s-shaped workload constraint
12 functions, denoted by Ω , a schedulability bound with respect to Ω is given by

$$13 \quad W^*(\theta) = \min_{\Gamma \in \Omega} \left(\min_{i=1, 2, \dots, n} \left(\sum_{\substack{j=1, 2, \dots, n \\ j \neq i}} \frac{F_j(\theta D_j)}{\theta D_j} + \min_{m=0, 1, \dots} \left(\frac{G_i(m \cdot S_i + D_i)}{\theta D_i + m \cdot \min\{\theta D_i, S_i\}} \right) \right) \right). \quad (\text{III.6})$$

14 By substituting (II.9) into (III.6), and then optimizing the resulting inequality, we have the

1 following closed-formed schedulability bound for the rate monotonic schedulers.

2

3 **Theorem 3:** A schedulability bound for the rate monotonic scheduler with s-shaped tasks is

4 given by

$$5 \quad W^*(1) = \begin{cases} \min \left(1, \frac{\eta}{k} \left(n \left(\left(\frac{\eta+1}{\eta} k \right)^{\frac{1}{n}} - 1 \right) + 1 - k \right) \right) & k \leq 1; \\ nk\eta \left(\left(\frac{k\eta+1}{k\eta} \right)^{\frac{1}{n}} - 1 \right) & k = 2, 3, \dots \end{cases} \quad (III.7)$$

6 where $\eta = V(\lceil k \rceil, \Gamma)$ and V is define defined in (III.4).

7 **Proof:** See [11].

8 (III.7) forms a three dimensional bound surface characterized by k , η and n . It is compatible

9 with existing known results for static priority systems, and represents a much broader range of

10 real-time systems. To make a side-by-side performance comparison between (III.7) and

11 previously known bounds, we set the scaling parameter θ defined in (II-5) to 1, and derive the

12 following relationship between workload rate and the classical utilization:

$$13 \quad W(1, \Gamma) = \sum_{i=1}^n \frac{F_i(kP_i)}{kP_i} = \frac{\lceil k \rceil}{k} \sum_{i=1}^n \frac{C_i}{P_i}. \quad (III.8)$$

14 The right hand side of (III.8) is obtained by the identity of $F_i(kP_i) = \lceil kP_i / P_i \rceil C_i$, according to

15 (II.6). For the heterogeneity function V defined in (III.4), we set the measuring window $\hbar = \lceil k \rceil$,

16 because this setting can simplify the derivation of the schedulability bound.

IV. Performance Evaluation and Comparison

On the basis of the results developed in the previous section, we evaluate the system performance in this section, using the schedulability bound as the primary performance measure. We will also show how existing schedulability bounds can be easily obtained from our result by a simple plug-in of proper system parameters.

A. Analysis of the Rate Monotonic Scheduler's Schedulability Bounds

By a close observation of Theorem 3, we notice that the following factors affect the schedulability bound:

- 1) *The normalized deadline, k .* When normalized deadlines of tasks become tighter, the expected schedulability bound decreases.
- 2) *The heterogeneity of a task set, η .* This parameter gauges the fitness of the workload constraint functions in capturing the spread of job sizes. It allows one to improve resource allocation without requiring the use of worst case job size in the periodic model. The system schedulability bound is expected to improve as η increases.

The sensitivity of the schedulability bound with respect to two key factors, k and η is analyzed, and the results are plotted in Figure 3. The relationship between the schedulability bounds and the two parameters are examined for a thousand tasks, or $n = 1,000$. For ease of comparison with existing results, we converted the workload rate to the classical utilization based on (III.8), and we varied k from 10^{-4} to 10^4 and η from 1 to 10^2 . The following observations can be made:

- 1) As expected, the tighter the deadlines, the lower the schedulability bounds. The sensitivity is especially significant when k is small (i.e., less than 10). For example, in Figure 3, when k changes from 10^{-1} to 10, the schedulability bound increases from 10.0% to 95.3%. The

1 sensitivity becomes less significant for large k values. In Figure 3, when k changes from 10

2 to 1,000, the schedulability bound increases from 95.3% to 99.9%, only a 4.6% gain.

3 2) Large heterogeneity leads to improved schedulability bounds. The sensitivity is high when η

4 is small, (i.e., 10), and becomes lower for larger η values. For example, in Figure 3, when k

5 and η changes from 1 to 10, the schedulability bound increases from 69.3% to 95.3%. When

6 η changes from 10 to 100, the schedulability bound increases from 95.3% to 99.5%, a mere

7 4.2% gain.

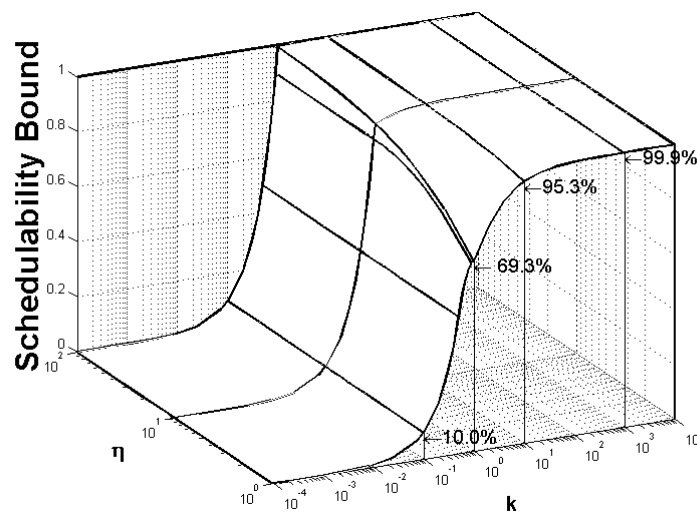


Fig. 3: Our New Schedulability Bounds Surface of the Rate Monotonic Scheduler

8 B. Parametric Fitting of Existing Utilization Bounds for the Rate Monotonic Scheduler

9 Theorem 3 gives the schedulability bounds for a wide range of systems based on three

10 parameters, i.e., normalized deadline, heterogeneity of task set, and number of tasks. It is much

11 more general than results reported in [3-8] that are merely one point and three curves in the 3-D

12 graph surface constructed from our bound result, as illustrated in Figure 4.

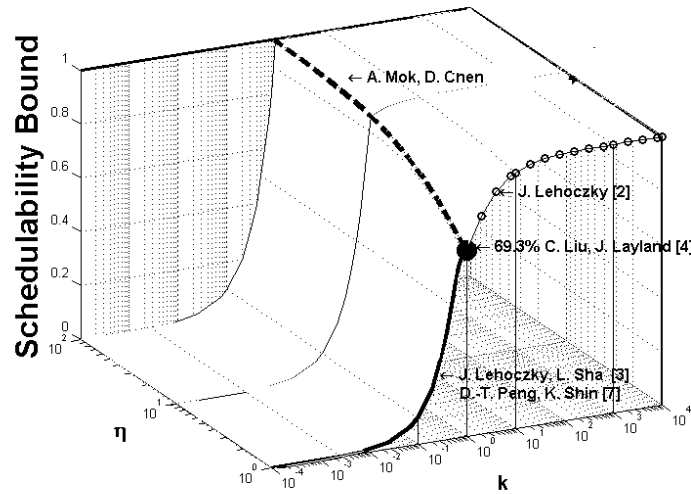


Fig. 4: Comparison of New Schedulability Bounds with Previous Results

1 In the following paragraphs, we illustrate how to reduce Theorem 3 into the bounds obtained
 2 in previous studies. Except for the slight difference between (IV.7) and (IV.8), we match all
 3 other results mentioned above.

- 4 • First, we consider the classical rate monotonic scheduler with periodic tasks, whose deadlines
 5 are equal to their periods. In [5], Liu and Layland derived a schedulability bound as follows:

$$6 \quad U^* = n \left(2^{\frac{1}{n}} - 1 \right). \quad (IV.1)$$

7 Now, we re-derive (IV.1) by using Theorem 3. Note that by (III.8), the workload rate reduces
 8 to the classical definition of utilization in this case. Since this is a periodic system, by (III.3)
 9 and (III.4), we have $\eta = V(1, \Gamma) = 1$. Because the relative deadline is equal to the length of
 10 period, we have $k = 1$. Substituting $\eta = 1$ and $k = 1$ into (III.7), we have

$$11 \quad U^* = W^*(1) = n \left(2^{\frac{1}{n}} - 1 \right). \quad (IV.2)$$

1 As n approaches infinity, U^* and W^* will approximately become 69.3%. This is represented
 2 by one single point in Figure 4 – which is superimposed on the Figure 3 to highlight the
 3 match.

4 • Second, we analyze the periodic system in which task's deadlines are shorter than periods,
 5 and tasks are scheduled by a rate monotonic scheduler. In [4] and [8], Lehoczky, Sha, Peng,
 6 and Shin derived the following utilization bound

$$7 \quad U^* = \begin{cases} k & k \leq \frac{1}{2} \\ n \left((2k)^{\frac{1}{n}} - 1 \right) + 1 - k & \frac{1}{2} < k \leq 1 \end{cases} . \quad (IV.3)$$

8 We now re-derive (IV.3) using Theorem 3. By (III.3) and (III.4), we have $\eta = 1$. Because the
 9 relative deadlines are less than the period lengths, we know that $k < 1$. Substituting $\eta = 1$ into
 10 (III.7), we get

$$11 \quad W^*(1) = \begin{cases} 1 & k \leq \frac{1}{2} \\ \frac{1}{k} \left(n \left((2k)^{\frac{1}{n}} - 1 \right) + 1 - k \right) & \frac{1}{2} < k \leq 1 \end{cases} \quad (IV.4)$$

12 However, by (III.8), in this case the workload rate and the classical utilization have the
 13 relationship

$$14 \quad W^*(1) = \frac{U^*}{k} . \quad (IV.5)$$

15 Thus, by substituting (IV.5) into (IV.4) and rearranging it, we have

$$U^* = kW^*(1) = \begin{cases} k & k \leq \frac{1}{2} \\ n \left((2k)^{\frac{1}{n}} - 1 \right) + 1 - k & \frac{1}{2} < k \leq 1 \end{cases} \quad (\text{IV.6})$$

This is exactly the same as (IV.3). We illustrate this result by a curve in the 3-D graph in Figure 4.

- Third, we analyze the periodic system in which task deadlines are multiples of periods and tasks are scheduled by a rate monotonic scheduler. In [3], Lehoczky obtained an utilization bound for $k = 2, 3, \dots$,

$$U^* = k(n-1) \left(\left(\frac{k+1}{k} \right)^{\frac{1}{n-1}} - 1 \right). \quad (\text{IV.7})$$

Note that by (III.8), the workload rate in this case reduces to the classical utilization. We will re-derive (IV.7) through Theorem 3. Since this is a periodic system, we have $\eta = 1$. Because the relative deadline is a multiple of period, we know that k is a positive integer greater than 1. By substituting $\eta = 1$ into (III.7), we get

$$U^* = W^*(1) = kn \left(\left(\frac{k+1}{k} \right)^{\frac{1}{n}} - 1 \right). \quad (\text{IV.8})$$

Note that (IV.8) is not as tight as (IV.7) (n vs. $n - 1$). This is due to the fact that Theorem 3 is obtained for general s-shaped functions. However, the exact bound of (IV.7) can be easily obtained with our general method. See [11] for this derivation. In Figure 4, a curve illustrates this match of results.

- 1 • Fourth, we analyze the AM multiframe system [6], [7] in which task's deadlines are equal to
 2 periods and tasks are scheduled by a rate monotonic scheduler. In [6] and [7], Mok and Chen
 3 obtained a utilization bound as follows

$$4 \quad U^* = rn \left(\left(\frac{r+1}{r} \right)^{\frac{1}{n}} - 1 \right), \quad (IV.9)$$

5 where $r = \min_{i=1, \dots, n} \{E_i^0 / E_i^1\}$, and $E_i^1 = C_i^2$, $E_i^0 = C_i^1$, are the size of the first and second jobs.

6 By (III.8), the workload rate is the same as the utilization defined in [6] and [7]. Now we re-
 7 derive (IV.9) from Theorem 3. Since this is a multiframe system, by (III.4), (III.3), we have η
 8 $= V(1, \Gamma) = \min_{i=1, \dots, n} \{C_i^2 / C_i^1\} = \min_{i=1, \dots, n} \{E_i^1 / E_i^0\} = r$. Because relative deadlines are equal
 9 to periods, we have $k = 1$. Substituting $\eta = r$, and $k = 1$ into (III.7), we get

$$10 \quad W^*(1) = rn \left(\left(\frac{r+1}{r} \right)^{\frac{1}{n}} - 1 \right), \quad (IV.10)$$

11 which is exactly the same as (IV.9). Again, this match is drawn on Figure 4.

12 Through the analysis mentioned above, we have illustrated that the results obtained in [3-8] are
 13 special cases of Theorem 3. Furthermore, our results cover many cases that have not been
 14 analyzed before, because the results from [3-8] are merely one point and three curves in the 3-D
 15 graph shown in Figure 4.

16 **V. Extensions**

1 A. Harmonic Chain Tasks

2 In [9] and [10], the authors proved that the utilization bound for periodic task improves when
3 the periods of tasks are divisible. This observation is applicable to Theorem 3 as stated below.

4

5 **Corollary 2:** Given the rate monotonic scheduler and a task set Γ with s-shaped workload
6 constraint functions, Γ is schedulable if

$$7 \quad W^*\left(\frac{1}{\lambda}\right) = \begin{cases} \min\left(1, \frac{\eta\lambda}{k} \left(n' \left(\frac{\eta+1}{\eta} \frac{k}{\lambda}\right)^{\frac{1}{n'}} - n'+1 - \frac{k}{\lambda}\right)\right) & \text{if } k \leq \lambda; \\ \frac{k}{\lambda} \eta n' \left(\left(\frac{k\eta + \lambda}{k\eta}\right)^{\frac{1}{n'}} - 1\right) & \text{if } k/\lambda = 1, 2, 3, \dots \end{cases} \quad (\text{V.1})$$

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

9 **Proof:** See [11].

10 B. Arbitrary Static Priority Schedulers

11 In this section, we extend Theorem 3 for arbitrary static priority schedulers. To model arbitrary
12 static priority schedulers, we introduce a new parameter, λ_i , the degree of deadline inversion for
13 task T_i , as follows,

$$14 \quad \lambda_i = \frac{\max_{j=1,2,\dots,i} (D_j)}{D_i}. \quad (\text{V.2})$$

15 Note that $\lambda_i \geq 1$. When $\lambda_i = 1$, deadlines of tasks with priority higher than T_i are less than or

1 equal to D_i . Hence, no deadline inversion occurs to T_i . When $\lambda_i > 1$, deadline inversion occurs
 2 to task T_i . Taking one step further, we define λ , the degree of deadline inversion for the whole
 3 task set as:

$$4 \quad \lambda = \max(\lambda_1, \lambda_2, \dots, \lambda_n) \quad (\text{V.3})$$

5 When $\lambda = 1$, the scheduler becomes deadline monotonic. Furthermore, when $\lambda = 1$ and $D_i = S_i$,
 6 the scheduler becomes rate monotonic.

7 **Corollary 3:** For an arbitrary static priority scheduler with s-shaped tasks, a service constraint
 8 function for task T_i is

$$9 \quad G_i(I) = \max_{0 \leq x \leq I} \left(x - \sum_{D_j \leq \lambda D_i} F_j(x) \right). \quad (\text{V.4})$$

10 **Proof:** By (V.2) and (II.8), we know Corollary 3 is true.

11 By substituting (V.4) into (V.1) and minimizing the subsequent inequality, we have the
 12 following results on the schedulability bound of arbitrary static priority scheduler with s-shaped
 13 tasks.

14 **Theorem 4:** Given a static priority scheduler and collection of tasks with s-shaped workload
 15 constraint functions, a schedulability bound is given by

$$16 \quad W^*\left(\frac{1}{\lambda}\right) = \begin{cases} \min\left(1, \frac{\lambda\eta}{k} \left(n \left(\frac{\eta+1}{\eta} \frac{k}{\lambda} \right)^{\frac{1}{n}} - n + 1 - \frac{k}{\lambda} \right) \right) & \text{if } k \leq \lambda; \\ \frac{k}{\lambda} \eta n \left(\left(\frac{k\eta + \lambda}{k\eta} \right)^{\frac{1}{n}} - 1 \right) & \text{if } k/\lambda = 1, 2, 3, \dots \end{cases} \quad (\text{V.5})$$

1 **Proof:** See [11].

2 **C. Non-preemptive Systems**

3 In a non-preemptive system, a high priority task T_i can be blocked by a lower priority task
 4 T_j , $j > i$, for a length of B_{\max} , the maximum job size of T_j . Thus, this non-preemptive effect
 5 will lead to a priority inversion interval for a length of B_{\max} . During this interval, the system acts
 6 as if T_j has a higher priority than T_i . In the worst case, the system in this interval is operating in
 7 a mode with $\lambda = \max_{i=1, 2, \dots, n} (D_i) / \min_{i=1, 2, \dots, n} (D_i)$. By substituting this into (V-5), we easily
 8 obtain the schedulability bound for non-preemption static priority systems.

9 **Corollary 4:** Given a non-preemptive static priority scheduler and a task set Γ with s-shaped
 10 workload constraint functions, Γ is schedulable if

$$11 \quad W^*\left(\frac{1}{\lambda^*}\right) = \begin{cases} \min\left(1, \frac{\eta\lambda^*}{k} \left(n\left(\frac{\eta+1}{\eta} \frac{k}{\lambda^*}\right)^{\frac{1}{\eta}} - n + 1 - \frac{k}{\lambda^*}\right)\right) & \text{if } k \leq \lambda^*; \\ \frac{k}{\lambda^*} \eta n \left(\left(\frac{k\eta + \lambda^*}{k\eta}\right)^{\frac{1}{\eta}} - 1\right) & \text{if } k / \lambda^* = 1, 2, \dots \end{cases} \quad (\text{V.6})$$

12 where $\lambda^* = \max_{i=1, 2, \dots, n} \{D_i\} / \min_{i=1, 2, \dots, n} \{D_i\}$ and n is the number tasks.

13 **Proof:** See [11].

14 **D. Utilization Bound of Rate Monotonic Schedulers with R-shaped Workload Constraint** 15 **Functions**

16 In this section, we will introduce an *r-shaped* workload constraint function and derive

1 schedulability bounds for static priority schedulers with this type of workload constraint
 2 function. We say a function $F(I)$ is *r-shaped* if $\forall s, t, 0 < s < t$, the following condition holds

$$3 \quad F(s)/s \geq F(t)/t. \quad (\text{V.7})$$

4 (V.7) means that the rate of the function $F(I)$ is not increasing with I and can be thought of
 5 as a special case of the s-shaped workload constraint when the segment length $S \rightarrow 0$. We call a
 6 task with r-shaped workload constraint function an r-shaped task. Compared with s-shaped
 7 functions, r-shaped function is much simpler and this simplicity can greatly reduce the
 8 complexity of the schedulability bound analysis as we will clarify.

9
 10 **Corollary 5:** Given a scheduler and collection of task set with r-shaped workload constraint
 11 functions, denoted by Ω , a schedulability bound with respect to Ω is given by

$$12 \quad W^*(\theta) = \inf_{\Gamma \in \Omega} \left(\min_{i=1, 2, \dots, n} \left(\inf_{I \geq 0} \left(\sum_{j=1, 2, \dots, n, j \neq i} \frac{F_j(\theta D_j)}{\theta D_j} + \frac{G_i(I + D_i)}{I + \theta D_i} \right) \right) \right). \quad (\text{V.8})$$

13 **Proof:** By Theorem 2 in [1], we know that a schedulability bound is

$$14 \quad W^*(\theta) = \inf_{\Gamma \in \Omega} \left(\sum_{j=1}^n \frac{F_j(\theta D_j)}{\theta D_j} + \min_{i=1, 2, \dots, n} \left((\alpha_i - 1) \frac{F_i(\theta D_i)}{\theta D_i} \right) \right), \quad (\text{V.9})$$

15 where

$$1 \quad \alpha_i = \inf_{I>0} (G_i(I+D) / F_i(I)) \quad (\text{V.10})$$

2 Since F_i is non-decreasing and is r-shaped, we have the following:

$$3 \quad \frac{F_i(\theta D_i)}{\theta D_i} \geq \frac{F_i(I + \theta D_i)}{I + \theta D_i} \geq \frac{F_i(I)}{I + \theta D_i} \quad (\text{V.11})$$

4 Thus,

$$5 \quad \frac{G_i(I+D)}{F_i(I)} \frac{F_i(\theta D_i)}{\theta D_i} \geq \frac{G_i(I+D)}{I + \theta D_i} \quad (\text{V.12})$$

6 By substituting (V.12) and (V.10) into (V.9) and re-arranging it, we have (V.8).

7 **Theorem 5:** Given a static priority scheduler and r-shaped tasks, a schedulability bound is

$$8 \quad W^*(1) = 1 / \lambda.$$

9 **Proof:** By Theorem 1, we know that a service constraint function for task T_i is

$$10 \quad G_i(I) = \max_{0 \leq x \leq I} \left(x - \sum_{j=1}^{i-1} F_j(x) \right). \quad (\text{V.13})$$

11 By (V.13) and Corollary 5, we have

$$12 \quad W^*(1) \geq \min_{i=1, 2, \dots, n} \left(\sum_{j=1}^{i-1} \frac{F_j(D_j)}{D_j} + \min_{I>0} \left(\frac{\max_{0 \leq x \leq I+D_i} \left(x - \sum_{j=1}^{i-1} F_j(x) \right)}{I + D_i} \right) \right). \quad (\text{V.14})$$

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

$$3 \quad \max_{0 \leq x \leq I+D_i} \left(x - \sum_{j=1}^{i-1} F_j(x) \right) \geq \max \left(0, I + D_i - \sum_{j=1}^{i-1} F_j(I + D_i) \right). \quad (\text{V.15})$$

4 By substituting (V.15) into (V.14), we have the following

$$5 \quad W^*(1) \geq \min_{i=1, 2, \dots, n} \left(\min_{I>0} \left(\sum_{j=1}^{i-1} \frac{F_j(D_j)}{D_j} + \frac{\max \left(0, I + D_i - \sum_{j=1}^{i-1} F_j(I + D_i) \right)}{I + D_i} \right) \right) \quad (\text{V.16})$$

6 By (V.7), we have

$$7 \quad \frac{F_j(D_i)}{D_i} \geq \frac{F_j(I + D_i)}{I + D_i}. \quad (\text{V.17})$$

8 By substituting (V.17) into (V.16), we have

$$9 \quad W^*(1) \geq \min_{i=1, 2, \dots, n} \left(\sum_{j=1}^{i-1} \frac{F_j(D_j)}{D_j} + \max \left(0, 1 - \sum_{j=1}^{i-1} \frac{F_j(D_i)}{D_i} \right) \right). \quad (\text{V.18})$$

10 From (V.2) and (V.3), we have

$$11 \quad D_i \geq D_j / \lambda. \quad (\text{V.19})$$

1 By substituting (V.19) into (V.7), we get

$$2 \quad \frac{F_j(D_i)}{D_i} \leq \frac{F_j(D_j/\lambda)}{D_j/\lambda}. \quad (\text{V.20})$$

3 Since $\lambda \geq 1$, we have $D_j/\lambda \leq D_j$ and thus

$$4 \quad \frac{F_j(D_i)}{D_i} \leq \frac{F_j(D_j)}{D_j/\lambda}. \quad (\text{V.21})$$

5 By substituting (V.21) into (V.18), we get

$$6 \quad W^*(1) \geq \min_{i=1,2,\dots,n} \left(\sum_{j=1}^{i-1} \frac{F_j(D_j)}{D_j} + \max \left(0, 1 - \sum_{j=1}^{i-1} \frac{\lambda F_j(D_j)}{D_j} \right) \right). \quad (\text{V.22})$$

7 Re-arranging (V.22) into

$$8 \quad W^*(1) \geq \min_{i=1,2,\dots,n} \left(\max \left(\sum_{j=1}^{i-1} \frac{F_j(D_j)}{D_j}, 1 - (\lambda - 1) \sum_{j=1}^{i-1} \frac{F_j(D_j)}{D_j} \right) \right). \quad (\text{V.23})$$

9 It can be verified the following

$$10 \quad \max \left(\sum_{j=1}^{i-1} \frac{F_j(D_j)}{D_j}, 1 - (\lambda - 1) \sum_{j=1}^{i-1} \frac{F_j(D_j)}{D_j} \right) \geq \frac{1}{\lambda}. \quad (\text{V.24})$$

1 Then, by substituting (V.24) into (V.23), we have

$$2 \quad W^*(1) \geq \min_{i=1, 2, \dots, n} (1/\lambda) = 1/\lambda. \quad (\text{V.25})$$

3 □

4 **VI. Final Remarks**

5 In this paper, we derive a closed-formed schedulability bound for arbitrary static priority
6 schedulers using the unified schedulability analysis framework introduced in [1]. The bound is
7 parameterized for different priority assignments and various workload patterns. The sensitivity
8 of system performance, i.e., schedulability bound, is analyzed with respect to normalized
9 deadline and heterogeneity of tasks. We show that our general schedulability bounds reduce to
10 the utilization bounds derived in [3-8] by simple plug-in of proper parameters.

11 The work reported in this paper can be extended in many different ways. This paper assumes
12 that the parameters of the tasks can be arbitrary and derives the worst-case schedulability bound
13 for the entire task set population. In practical systems, it is common for the task collection to be
14 finite and some of the parameter selection is subject to predefined constraint. For example, the
15 length of the segments S_i can be only chosen from a finite set $\{S_1, S_2, S_3, \dots, S_m\}$. In these cases,
16 the schedulability bound can be further improved.

17 **Acknowledgment**

18 This work was supported in part by the National Science Foundation under Contracts 0081761,

1 0324988, 0329181, 0530210, 0516825, by the Defense Advanced Research Projects Agency
2 under Contract F30602-99-1-0531, and by Texas A&M University under its
3 Telecommunications and Informatics Task Force Program. Any opinions, findings, conclusions,
4 and/or recommendations expressed in this material, either expressed or implied, are those of the
5 authors and do not necessarily reflect the views of the sponsors listed above. In addition, we are
6 grateful to Larisa Archer for helping us to edit the entire manuscript.

1 References

- 2 [1] Jianjia Wu, Jyh-Charn Liu, and Wei Zhao , “A General Framework for Parameterized
3 Schedulability Bound Analysis of Real-time Systems” , Technical Report 2008-3-3,
4 Computer Science Department, Texas A&M University
- 5 [2] J. Wu, J.-C. Liu and W. Zhao, “On schedulability bounds of static priority schedulers,”
6 IEEE Real-Time and Embedded Technology and Applications Symposium, San Francisco,
7 California, pp. 529-540, Mar. 2005.
- 8 [3] J. P. Lehoczky, “Fixed priority scheduling of periodic task sets with arbitrary deadlines,”
9 *Proc. Eleventh IEEE Int. Real Time Systems Symposium*, Piscataway, New Jersey, pp. 201-
10 209, Dec. 1990.
- 11 [4] J. P. Lehoczky and L. Sha, “Performance of real-time bus scheduling algorithms,” *ACM*
12 *SIGMETRICS Performance Evaluation Review*, vol. 14, no. 1, pp. 44–53, May 1986.
- 13 [5] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-
14 time environment,” *J. ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.
- 15 [6] A. K. Mok and D. Chen, “A general model for real-time tasks,” Technical Report TR-96-
16 24, Dept. Computer Sciences, Univ. Texas at Austin, Austin, Texas, Oct, 1996.
- 17 [7] A. K. Mok and D. Chen, “A multiframe model for real-time tasks,” *IEEE Trans. Software*
18 *Engineering*, vol.23, no.10, pp. 635-645, Oct. 1997.
- 19 [8] D.-T. Peng and K.G. Shin, “A new performance measure for scheduling independent real-
20 time tasks,” *J. Parallel Distributing Computing*, vol. 19, no. 12, pp. 11–16, Sept. 1993.
- 21 [9] D. Chen, A. K. Mok, and T.-W. Kuo, “Utilization bound revisited,” *IEEE Trans.*
22 *Computers*, vol. 52, no. 3, pp. 351- 361, Mar. 2003.
- 23 [10] T.-W. Kuo and A. K. Mok, "Load adjustment in adaptive real-time systems," *Proc. IEEE*

- 1 *Int. Real-Time Systems Symposium*, San Antonio, Texas, pp. 160-171, 1991.
- 2 [11] J. Wu, "General schedulability bound Analysis and its applications in real-time systems",
- 3 <http://txspace.tamu.edu/handle/1969.1/5854>, Ph.D. dissertation, Department of Computer
- 4 Science, Texas A&M University, College Station, Texas, 2006.