Heuristic Scheduling for Multimedia Streams with Firm Deadlines

Jinsung Cho and Heonshik Shin
Department of Computer Engineering
Seoul National University
Seoul 151-742, Korea
{cjs,shinhs}@comp.snu.ac.kr

Abstract

Multimedia streams such as audio and video require real-time manipulation. Unlike hard real-time tasks, however, they have firm deadlines since the loss of a few frames does not lead to a significant degradation in quality. In this paper, we propose a simple but efficient scheduling scheme for multimedia streams using heuristic functions. The proposed heuristic function takes into account the fail ratio and the consecutive fail count in order to avoid consecutive fails of a multimedia task, and at the same time, to meet the given quality of all the accepted multimedia tasks in the system. By gracefully degrading the quality of the existing multimedia tasks, our scheme can meet the QoS of a newly arriving task accepted by the proposed admission control criteria. Extensive simulation has been performed to validate the effectiveness of the scheme under various multimedia task sets and QoS parameters.

1. Introduction

Past researches on real-time scheduling theory have focused on the scheduling policy and its schedulability analysis under hard real-time constraints. The static scheduling policy represented by rate monotonic(RM) and the dynamic scheduling by earliest deadline first(EDF) have been extended to accommodate more realistic assumptions than those given by Liu and Layland[6]([8, 13]), and the analysis of algorithms for periodic tasks has been performed to incorporate sporadic or aperiodic tasks[10, 2]. The real-time tasks in these works have *hard* deadlines of which the miss can be catastrophic.

In contrast, in a multimedia application which is a new area of real-time computing, it is not necessary for every instance of a repetitive task to meet its deadline. This is because the loss of a few frames in multimedia streams does not cause a significant QoS degradation. So multimedia streams may have *firm* deadlines[4]. A deadline is said to be firm if the results produced by the corresponding task cease to be useful as soon as the deadline expires, but consequences of missing the deadline are not very severe[9]. On the other hand, deadline misses or failures of task execution may arise in high utilization or system overload. On arrival of a new stream in such a case, the quality of each stream which has been already accepted can be gracefully degraded to its minimum QoS value. Therefore, we should schedule multimedia streams to meet the QoS of all the accepted streams in the system, while avoiding consecutive misses of a stream which lead to a significant degradation in the quality.

In this paper, we propose a scheduling scheme for a set of multimedia streams which have firm deadlines and contain QoS parameters such as the minimum success ratio and the maximum consecutive fail count. The proposed scheme exploits a simple heuristic function considered for meeting the given QoS. Using extensive simulation, we select the proper parameters of the hueristic function and validate its effectiveness under various multimedia task sets and QoS parameters.

So far there have been a number of real-time task models proposed for multimedia streams. In [12], Tindell, *et. al.*, applied their fixed priority preemptive scheduling theory to multimedia disk traffic for the guaranteed retrieval. However, the high preemption cost of disk operations may prevent their model from effective implementation. As a similar approach, for a limited task set with the same execution time or the same period, Baek presented a non-preemptive schedulability analysis for the retrieval from disks[1]. Recently, Mok has proposed a new multiframe model for the task of which the execution time varies from one instance to another, and applied it to multimedia streams[7]. As mentioned earlier, however, every instance of multimedia streams may not meet its deadline. In [5], an integrated scheduling of multimedia and hard real-time tasks has been given but the QoS negotiation is accomplished by the multimedia server ratio (fixed computation time/period). There are two recent works closely related to ours under the sim-

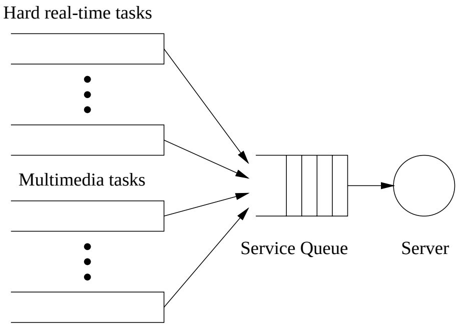


Figure 1. The system model. *The server executes a task with the highest priority in the service queue. All the hard real-time tasks have higher priority than multimedia tasks with firm deadlines.*

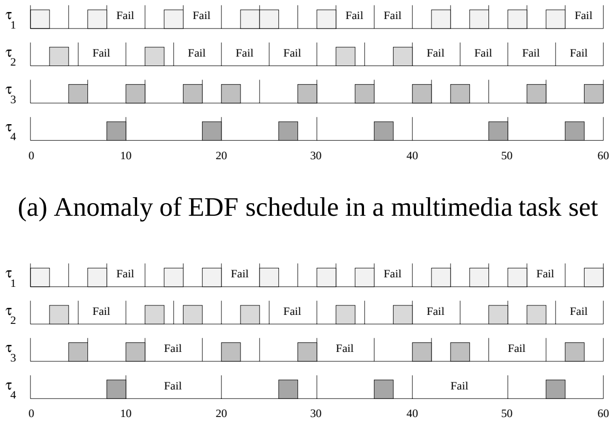
ilar system model[3, 4]. The detailed differences will be described later.

The rest of this paper is organized as follows: Section 2 describes the system model and the problems in scheduling multimedia streams, and we present the proposed scheduling scheme in Section 3. The empirical evaluation of our scheme is given in Section 4 and Section 5 concludes this paper.

2. System model and problem description

We consider a system with a single server, n multimedia tasks, and m hard real-time tasks as shown in Figure 1. In this generic service model, the server executes a task with the highest priority in the service queue, or among the released tasks. All the hard real-time tasks have higher priority than multimedia tasks with firm deadlines. A hard real-time task set is defined as $T_R = \{\tau_i^R \mid \tau_i^R = (C_i, P_i),\ i = 1, \cdots, m\}$ and a multimedia task set as $T_M = \{\tau_i^M \mid \tau_i^M = (r_i, C_i, P_i, Q_i, F_i),\ i = 1, \cdots, n\}$, where C_i denotes the computation time, P_i is the period, r_i is the release time, Q_i is the minimum success ratio, and F_i is the maximum consecutive fail count. The QoS of a multimedia task τ_i^M is characterized by Q_i and F_i. In other words, τ_i^M should be scheduled to meet Q_i success ratio while avoiding F_i consecutive fails. An instance of multimedia task, denoted as $\tau_{i,j}$, may consist of one or more frames. In addition, the current compression technique for multimedia streams enforces C_i to be variable.

This system and task model can be applied to real-time applications such as automated manufacturing and attack helicopters designed to take advantage of audio and video information[5]. Another applications include a multimedia server which transmits multiple streams to many clients and an end-system of video conferencing which displays multiple streams concurrently.



(a) Anomaly of EDF schedule in a multimedia task set

(b) Desired schedule

Figure 2. EDF schedule vs. desired schedule. *The EDF schedule of (a) is undesirable since τ_1 and τ_2 have many and consecutive fails. By gracefully degrading the QoS of τ_3 and τ_4, the acceptable schedule for τ_1 and τ_2 can be obtained like (b).*

In the system model given above, a straightforward approach for meeting the deadlines of tasks is to assign their priorities by EDF policy. The EDF scheduling is known to be optimal in the sense that it can meet all the deadlines for a task set of which the utilization is less than or equal to one[6]. However, when the utilization is more than one, or during system overload, the EDF policy generates an unacceptable schedule for a multimedia task set where the QoS of tasks can be degraded. Consider a simple task set $T_M = \{\tau_1 = (2, 4), \tau_2 = (2, 5), \tau_3 = (2, 6), \tau_4 = (2, 10)\}$, where a task is represented by (C_i, P_i). As shown in Figure 2(a), while the execution of τ_3 and τ_4 all succeed, many instances of τ_1 and τ_2 fail. It is also undesirable that consecutive fails occur, which causes a significant degradation in the quality of multimedia streams. The schedule like Figure 2(b) is acceptable since no consecutive fail occurs and the success ratios of the tasks are uniform.

In order to solve a similar problem, Hamdaoui[3] introduced the notion of (m, k)-firm deadlines. A stream is said to have (m, k)-firm deadlines if at least m out of any k consecutive instances must meet their deadlines, and to experience a *dynamic failure* if fewer than m out of any k consecutive instances meet their deadlines. They proposed a distance-based priority assignment technique to reduce the probability of dynamic failures, but did not consider consecutive frame misses. In their model, it is acceptable if $k - m$ consecutive frames miss their deadlines. However, it may cause a serious degradation in the quality of reconstructed pictures. In [4], by raising the priority of the urgent frames and pre-scheduling them with the backwards-EDF algorithm, the urgent frames can meet their deadlines and the normal frames have more room for their execution. The QoS parameter of their task model is given by K which is the minimum tolerable interval between two frame deadline misses. This may be too strict and the complexity of their algorithm is very high. In this paper, we intend to propose an efficient scheduling scheme that is less complex to implement and shows similar or better performance.

3. The proposed scheme

In order to gracefully degrade the QoS of all tasks, as shown in Figure 2(b), the priority of $\tau_{2,4}$(the forth instance of τ_2) should be given higher than that of $\tau_{3,3}$ although the deadline of $\tau_{3,3}$ is earlier than that of $\tau_{2,4}$. Similarly, $\tau_{2,10}$ should have higher priority than $\tau_{4,5}$ to avoid consecutive fails. The basic idea behind our scheme is to distribute the fails across all the tasks while meeting the given QoS of each task. We first present the following priority assignment scheme:

- assigns the priority to hard real-time tasks based on EDF policy to be higher than that of multimedia tasks,
- assigns the priority to multimedia tasks using a heuristic function $H(\tau)$,
- assigns the priority to multimedia tasks with the same heuristic value by EDF policy, and
- does not abort the task being executed.

First of all, the heuristic function $H(\tau)$ should be designed to avoid consecutive fails. So it should contain a factor to be considered for the consecutive fail counts with respect to the maximum count F of task τ. Next, $H(\tau)$ should take into account the success ratio with respect to minimum ratio Q. Considering these factors, we define the heuristic function $H(\tau)$ as follows:

$$H(\tau) = aR^{\tau}_{fail} + bR^{\tau}_{consecutive\_fail} + cR^{\tau}_{importance}, \quad (1)$$

where, for task τ, $R^{\tau}_{fail} = \frac{\text{fail ratio}}{1-Q}$[1], $R^{\tau}_{consecutive\_fail} = \frac{\text{consecutive fail count}}{F}$, and $R^{\tau}_{importance}$ is the degree of importance of task τ ranging from 0 to 1. The larger value of $H(\tau)$ represents higher priority. In general, $H(\tau)$ assigns higher priority to the tasks of which the previous instance failed, because $R^{\tau}_{consecutive\_fail} \neq 0$. The tasks for which $R^{\tau}_{consecutive\_fail} = 0$ have their priorities according to R^{τ}_{fail}, indicating the relative urgency determined on the basis of the QoS. When some tasks have the same heuristic value, we adopt the EDF policy considered for the high success ratio. In addition, the uncompleted tasks are not canceled, since aborting them wastes the computation time of the server by their usage and may lead to undesired results, so called the domino effect[11], in which a missed deadline causes a series of subsequent deadlines to be also missed. Consequently, the proposed scheme is expected to meet the QoS of multimedia tasks with firm deadlines and show the high server utilization. In the next section, we select the proper parameters of $H(\tau)$ and analyze the performance of the proposed scheme by simulation.

On the other hand, the QoS of multimedia tasks can be guaranteed by the following admission control criteria:

$$\sum_{i=1}^{m} U_i^R (= \frac{C_i}{P_i}) + \sum_{i=1}^{n} U_i^M (= \frac{C_i}{P_i} \cdot Q_i) \leq 1. \quad (2)$$

When a multimedia task which satisfies Eq. (2) arrives, in our scheme, by gracefully degrading the QoS of existing multimedia tasks, all the tasks in the multimedia task set can meet the QoS. If the computation time C_i of a multimedia task τ_i is given by the worst case value in Eq. (2), the deterministic guarantee can be achieved. The average value of C_i may guarantee the QoS statistically.

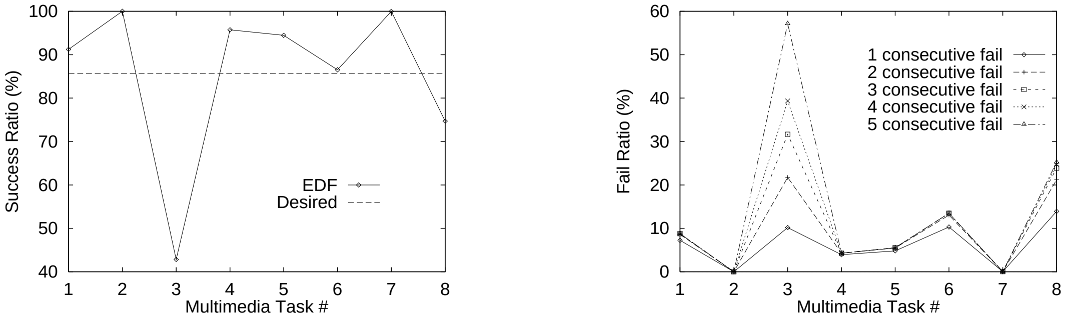
4. Performance evaluation

First, we begin by presenting the problems of the EDF scheduling described in Section 2. Figure 3 depicts the result for a homogeneous multimedia task set where the periods are all the same, that is, $P_i = 33$ msec, for $1 \leq i \leq n$ (30 frames/sec). The computation time of each task is uniformly distributed between 2 and 8 msec. As mentioned earlier, we can find that the tasks, τ_3 and τ_8 in Figure 3, have low success ratio and many consecutive fails.

To alleviate this untoward effect of EDF scheduling, we exploit a heuristic function $H(\tau)$ given in Eq. (1). For the purpose of selecting the parameters of $H(\tau)$, we set $a = \alpha$, $b = (1 - \alpha)$, and $c = 0$, because $R^{\tau}_{importance}$ is the application-dependent term. Varying α from 0 to 1, we carried out the extensive simulation with various multimedia task sets. Figure 4 shows one of the results for a heterogeneous multimedia task set where the periods of the tasks are different from each other. For this experiment, we choose a multimedia task set where the average utilization($\sum_{i=1}^{n} \frac{C_i^{avg}}{P_i} \cdot Q_i$) is close to one[2] and the QoS parameters are all the same, that is, $Q_i = 0.8$, $F_i = 3$, for $1 \leq i \leq n$. Hence, the similar results should be presented for all the tasks. As shown in Figure 4, however, when $H(\tau)$ considers only the consecutive fail ratio, that is, $\alpha = 0$, the success ratio of the tasks varies. Specifically, τ_6, τ_9, and τ_{11} do not meet the minimum success ratio of 80%. In contrast

[1] By the fail ratio, we mean (number of fail instances / total number of instances). In the rest of this paper, the term *ratio* is defined similarly.

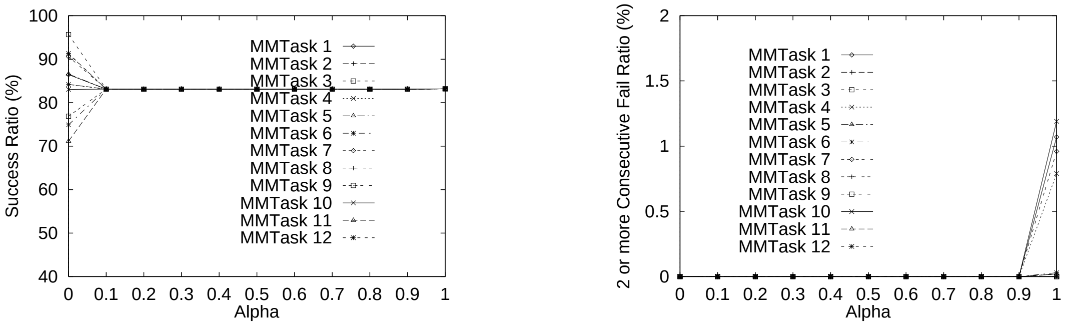
[2] The average utilization given in Figure 4 is 0.97.



(a) Success ratio

(b) Fail ratio

Figure 3. Problems of EDF scheduling for a homogeneous multimedia task set. *τ_3 and τ_8 show the unacceptable behavior.*



(a) Success ratio

(b) 2 or more consecutive fail ratio

Figure 4. Effect of α in $H(\tau)$ on performance for a heterogeneous multimedia task set with the same QoS parameter. *$P_{3i-2} = 33$ msec, $P_{3i-1} = 50$ msec, $P_{3i} = 100$ msec$(1 \leq i \leq 4)$. $C_i = 2 \sim 8$ msec, $Q_i = 0.8$, $F_i = 3(1 \leq i \leq 12)$. Success ratio and consecutive fail ratio are defined with respect to the total number of instances.*

to this case, when $\alpha = 1$, the tasks have met consecutive fails, because $H(\tau)$ does not take account of them. When $H(\tau)$ considers both terms, that is, $0 < \alpha < 1$, we can find the desired results. Based on these we choose $\alpha = 0.5$ for $H(\tau)$.

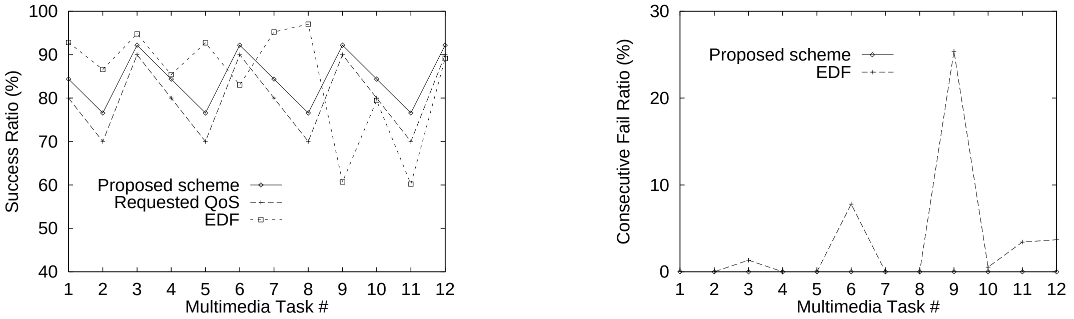
Next, the simulation proceeds to check whether our scheme can meet the various QoS for heterogeneous multimedia streams. Under the heavy load where the average utilization for the task set is 0.95, the proposed scheme can satisfy the requested QoS for all the tasks, as shown in Figure 5, while the EDF policy cannot.

Figure 6 shows the performance of our scheme for the case where hard real-time tasks and multimedia tasks are mixed. The hard real-time task set has the utilization of 0.5, and thus, the number of multimedia tasks decreased compared with the case in Figure 5. Since the periods of the hard real-time tasks are made short relative to those of the multimedia tasks, the hard real-time tasks will preempt the multimedia tasks frequently. However, the proposed scheme meet the requested QoS of multimedia tasks as shown in Figure 6.

We now compare the performance of our scheme with that of the distance-based priority(DBP) scheme proposed in [3]. As mentioned earlier, the QoS parameter in [3] is given by (m, k), while ours by (Q, F). In our scheme, Q can be modeled as m/k but lacks the notion of dynamic failure which is said to occur if fewer than m out of any k consecutive instances meet their deadlines[3]. Figure 7(a)[3] shows that the tasks experience dynamic failures although their presence is not integrated in our method.

On the other hand, the DBP scheme does not consider consecutive fails. In Figure 7(a), we can find that the DBP

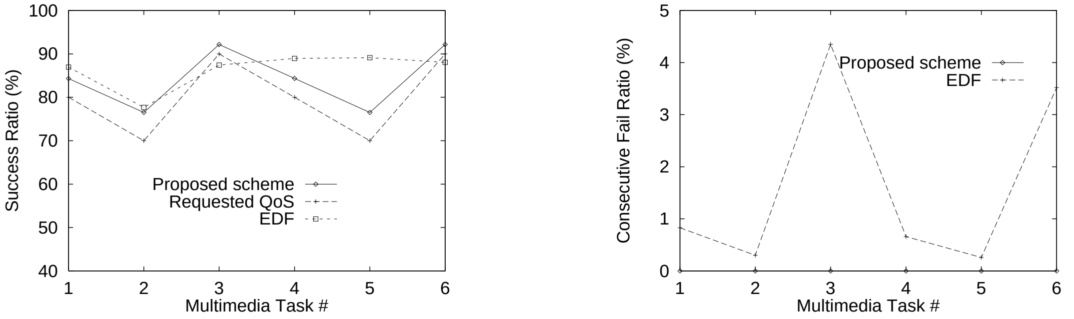
[3] In order to define dynamic failure, in our scheme, (m, k) is given by $(10Q, 10)$.



(a) Success ratio

(b) Not allowed consecutive fail ratio

Figure 5. Performance of the proposed scheme for a heterogeneous multimedia task set with the various QoS parameters. $\alpha = 0.5$. $P_{3i-2} = 33$ *msec*, $P_{3i-1} = 50$ *msec*, $P_{3i} = 100$ *msec*, $Q_{3i-2} = 0.8$, $Q_{3i-1} = 0.7$, $Q_{3i} = 0.9$, $F_{3i-2} = 2$, $F_{3i-1} = 3$, $F_{3i} = 1 (1 \leq i \leq 4)$. $C_i = 2 \sim 8$ *msec*$(1 \leq i \leq 12)$.



(a) Success ratio

(b) Not allowed consecutive fail ratio

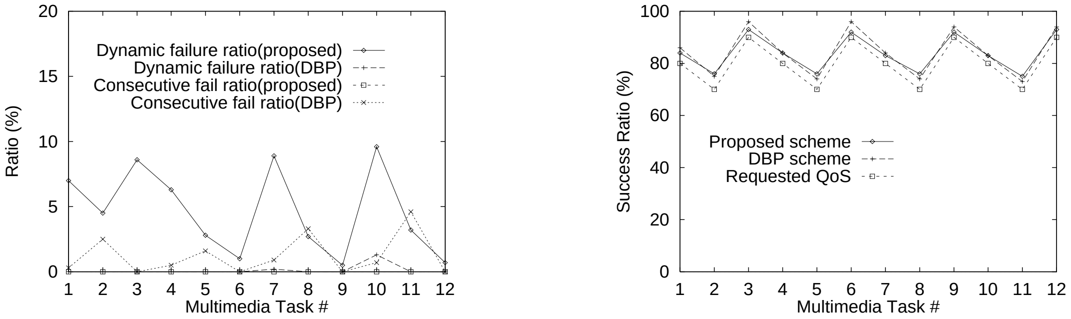
Figure 6. Performance of the proposed scheme for a multimedia task set with a hard real-time task set. $\alpha = 0.5$. $P_{3i-2} = 33$ *msec*, $P_{3i-1} = 50$ *msec*, $P_{3i} = 100$ *msec*, $Q_{3i-2} = 0.8$, $Q_{3i-1} = 0.7$, $Q_{3i} = 0.9$, $F_{3i-2} = 2$, $F_{3i-1} = 3$, $F_{3i} = 1 (1 \leq i \leq 2)$. $C_i = 2 \sim 8$ *msec*$(1 \leq i \leq 6)$. *The utilization of the hard real-time task set* T_R *is 0.5 where* $T_R = \{(2, 8), (0.9, 6), (0.5, 5)\}$.

scheme does not satisfy the maximum consecutive fail. In the case of success ratio, Figure 7(b) indicates that both of the schemes meet the minimum success ratio of each task with little difference. Consequently, they differ in whether they depend on dynamic failure or consecutive fail to effectively schedule multimedia streams. In other words, the same level of QoS may be maintained regardless of whether we prevent the incoming streams from dynamic failure or consecutive fail. Whether the system needs to improve dynamic failure ratio or consecutive fail ratio depends on the attributes of an application.

Since our scheme exploits the heuristic function, however, $H(\tau)$ can include additional factor, that is, dR^{τ}_{DBP} that accounts for the dynamic failure. In another experiments, we have found that, if $H(\tau)$ includes dR^{τ}_{DBP}, the tradeoff between dynamic failure and consecutive fail can be adjusted. Therefore, with high flexibility, we may adapt the proposed scheduling method to the requirement of applications by improving the heuristic function.

5. Conclusions

In this paper, we have proposed a simple but efficient scheduling scheme for multimedia tasks using heuristic functions. Since it is not necessary for every instance of a multimedia task to meet its deadline, a multimedia task is said to have firm deadline. The basic idea is to avoid consecutive frame misses of a stream, and at the same time, to meet the requested qualiy of all the accepted streams in the system. The proposed heuristic function considers the fail ratio



(a) Fail ratio

(b) Success ratio

Figure 7. Performance comparison with DBP scheme. *For the parameters of DBP scheme,* $m_{3i-2} = 8$, $m_{3i-1} = 7$, $m_{3i} = 9$, $k_j = 10(1 \leq i \leq 4,\ 1 \leq j \leq 12)$. $\alpha = 0.5$. $P_{3i-2} = 33$ *msec,* $P_{3i-1} = 50$ *msec,* $P_{3i} = 100$ *msec,* $Q_{3i-2} = 0.8$, $Q_{3i-1} = 0.7$, $Q_{3i} = 0.9$, $F_j = 1(1 \leq i \leq 4,\ 1 \leq j \leq 12)$. $C_i = 2 \sim 8$ *msec*$(1 \leq i \leq 12)$. *Dynamic failure ratio is defined with respect to the total number of instances.*

and the consecutive fail count for each multimedia task. By gracefully degrading the quality of the existing multimedia tasks, our scheme can meet the QoS of a newly arriving task accepted by the proposed admission control criteria. The simulation results show that the proposed scheme satisfies the requested QoS of multimedia tasks even with a hard real-time task set. The heuristic function may be modified depending on applications so that the proposed scheme also accounts for the dynamic failures of incoming multimedia streams. This will render flexibility when different multimedia applications demand more appropriate scheduling schemes for their own effectiveness.

References

[1] Y. C. Baek and K. Koh. Schedulability analysis of non-preemptive real-time tasks for continuous media retrieval. *Proc. of the IEEE Tencon'94*, 1994.

[2] H. Chetto and M. Chetto. Some results of the earliest deadline scheduling algorithm. *IEEE Transactions on Software Engineering*, 15(10):1261–1269, 1989.

[3] M. Hamdaoui and P. Ramanathan. A dynamic priority assignment technique for streams with $(m,\ k)$-firm deadlines. *IEEE Transactions on Computers*, 44(12):1443–1451, 1995.

[4] C.-C. Han and K. G. Shin. Scheduling mpeg-compressed video streams with firm deadline constraints. *Proc. of ACM Multimedia 95*, pages 411–422, 1995.

[5] H. Kaneko and J. A. Stankovic. Integrated scheduling of multimedia and hard real-time tasks. *Proc. of the 17th Real-Time Systems Symposium*, pages 206–217, 1996.

[6] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of ACM*, 20(1):46–61, 1973.

[7] A. K. Mok and D. Chen. A multiframe model for real-time tasks. *Proc. of the 17th Real-Time Systems Symposium*, pages 22–29, 1996.

[8] L. Sha, R. Rajkumar, and S. S. Sathaye. Generalized rate-monotonic scheduling theory: A framework for developing real-time systems. *Proc. of the IEEE*, 82(1):68–82, 1994.

[9] K. G. Shin and P. Ramanathan. Real-time computing: A new discipline of computer science and engineering. *Proc. of the IEEE*, 82(1):6–24, 1994.

[10] B. Sprunt, L. Sha, and J. Lehoczky. Aperiodic task scheduling for hard-real-time systems. *Real-Time Systems*, 1(1):27–60, 1989.

[11] M. Spuri, G. Buttazzo, and F. Sensini. Robust aperiodic scheduling under dynamic priority systems. *Proc. of the 16th Real-Time Systems Symposium*, pages 210–219, 1995.

[12] K. Tindell, A. Burns, and R. Davis. Fixed priority scheduling of hard real-time multimedia disk traffic. *Proc. of the Workshop on the Role of Real-Time in Multimedia/Interactive Computing Systems*, 1993.

[13] K. W. Tindell, A. Burns, and A. J. Wellings. An extendible approach for analyzing fixed priority hard real-time tasks. *Real-Time Systems*, 6(2):133–151, 1994.