Robustness Results for Hierarchical Diff-EDF Scheduling upon Heterogeneous Real-Time Packet Networks

Moutaz Saleh and Zulaiha Ali Othman

Faculty of Science and IT, University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor

Abstract: Packet networks are currently enabling the integration of traffic with a wide range of characteristics that extend from video traffic with stringent QoS requirements to the best-effort traffic requiring no guarantees. QoS guarantees can be provided in conventional packet networks by the use of proper packet scheduling algorithms. As a computer revolution, many scheduling algorithms have been proposed to provide different schemes of QoS guarantees with Earliest Deadline First (EDF) as the most popular one. With EDF scheduling, all flows receive the same miss rate regardless of their traffic characteristics and deadlines. This makes the standard EDF algorithm unsuitable for situations in which the different flows have different miss rate requirements since in order to meet all miss rate requirements it is necessary to limit admissions so as to satisfy the flow with the most stringent miss rate requirements. In this study, we propose a new priority assignment scheduling algorithm, Hierarchal Diff-EDF (Differentiate Earliest Deadline First), which can meet the real-time needs of these applications while continuing to provide best effort service to non-real time traffic. The Hierarchal Diff-EDF features a feedback control mechanism that detects overload conditions and modifies packet priority assignments accordingly.

Key words: Robustness results, integration of traffic, heterogeneous real-time packet networks

INTRODUCTION

Recently, many applications of computer networks rely on the ability of the network to provide Quality of Service (QoS) guarantees. These guarantees are usually bounded in the form of delay, bandwidth, packet loss rate and buffer utilization or a combination of these parameters. Furthermore, packet networks are currently enabling the integration of traffic with a wide range of characteristics that extend from video traffic with stringent QoS requirements to ``best-effort'' traffic requiring no guarantees. QoS guarantees can be provided in conventional packet networks by the use of proper packet scheduling algorithms. The function of a scheduling algorithm is to select the packet to be transmitted in the next cycle from the available arrived packets.

Network traffic can be categorized into two types: real-time traffic, such as video and audio and non-real-time traffic such as http data. Recently, there has been a significant increase in the amount of multimedia services transmitted over networks. These multimedia applications, due to the stringent delay constraints, have to meet certain QoS guarantees. Since scheduling has a direct impact on the system capacity and delay as well

as throughput, it is therefore necessary to investigate the suitable scheduling algorithms for multimedia traffic.

The distinguishing characteristic of real-time traffic is that it requires bounded delay while it can tolerates some packet losses. The delay can be bounded by associating a deadline for each packet. Once a packet misses its deadline, it will be dropped as it is no longer useful. Therefore the main goal for any scheduling scheme for real-time traffic is to deliver packets in a timely manner.

As a computer revolution, many scheduling algorithms have been proposed to meet this goal. The First-Come-First-Serve (FCFS) scheduling algorithm, which is mostly used in conventional networks, is widely adopted for best-effort traffic. On the other hand, many scheduling algorithms have been proposed to provide different schemes of QoS guarantees, with Earliest Deadline First (EDF) as the most popular one.

Real-time systems: A real-time system has two notions of correctness: logical and temporal[1]. In particular, in addition to producing correct outputs (logical correctness), such a system needs to ensure that these outputs are produced at the correct time (temporal

Corresponding Author: Moutaz Saleh, Faculty of Science and IT, University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor

correctness). However, selecting appropriate methods for scheduling activities is one of the important considerations in the design of a real-time system[2]; such methods are essential to ensure that all activities are able to meet their timing constraints. These timing constraints are usually specified using a deadline, which corresponds to the time by which a specific operation must complete.

Real-time systems can be broadly classified as hard or soft depending on the criticality of deadlines[3]. In hard real-time systems, all deadlines must be met; equivalently, a deadline miss results in an incorrect system. On the other hand, in a soft real-time system, timing constraints are less stringent; occasional deadline misses do not affect the correctness of the system.

A real-time system is typically composed of several or sequential tasks with timing constraints. In most real time systems, tasks are invoked repeatedly: each invocation of a task is referred as a job; and the corresponding time of invocation is referred as the job's release time or job's deadline[1]. Thus, the relative deadline parameter is used to specify the timing constraints of the jobs.

Related work: Many real-time systems rely on the earliest deadline first (EDF) scheduling algorithm. This algorithm has been shown to be optimal under many different conditions. For example, for independent, preemptable tasks, on uni-processor EDF is optimal in the sense that if any algorithm can find a schedule where all tasks meet their deadline then EDF can meet the deadline[4]. Also, Jackson's rule[5] says that ordering a set of tasks by deadline will minimize the maximum lateness. Further, it also has been shown that EDF is optimal under certain stochastic conditions[6].

In spite of these advantageous properties, EDF has one major negative aspect. That is, when using EDF in a dynamic system, if overload occurs, tasks may miss deadlines in an unpredictable manner and in the worst case, the performance of the system can approach zero effective throughput. This is due to the fact that EDF gives highest priority to those processes that are close to missing their deadlines. In such situations, EDF does not provide any type of guarantee on which tasks will meet their timing constrains. This is a very undesirable behavior in practical systems, since in real-world applications intermittent overloads may occur due to exceptional situations, such as modifications in the environment, arrival of a burst of tasks, or cascades of system failures. In that case, matters may be improved by introducing some congestion control mechanism.

A robust earliest deadline scheduling algorithm for dealing with sporadic tasks under overloads in hard real-time environment was proposed by[7]. The algorithm synergistically combines many features including a very minimum level of guarantee, dynamic guarantees, graceful degradation in overloads, deadline tolerance and resource reclaiming. Also, in 1995[8] presented a comparative study among scheduling algorithms, which use different priority, assignments and different guarantee mechanisms to improve the performance of a real-time system during overload conditions. Their results showed that EDF scheduling performs best if admission control is used along with a reclaiming mechanism that's takes advantage of early completions. In 1997[9] introduced algorithms for flow admission control at an EDF link scheduler. Their results showed that these algorithms have very low computational complexity and are easily applicable in practice.

While real-time system designers try to design the system with sufficient resources, because of cost and highly unpredictable environments, it is sometimes impossible to guarantee that the system resources are sufficient; in this case EDF's performance degrades rapidly in overload situations. However, it is worthy to say that in the year of 1998, EDF was a major paradigm for real-time scheduling[10].

EDF is a widely used algorithm for online deadline scheduling. It has been known for long that EDF is optimal for scheduling an underloaded, single processor system; recent results on the extra-resource analysis of EDF further revealed that EDF when using moderately faster processors can achieve optimal performance in the under loaded, multi-processor setting[11], initiated the extra resource analysis of EDF for overloaded systems, showing that EDF supplemented with simple form of admission control can provide a similar performance guarantee in both single and multi-processor settings.

Also, EDF is widely used in scheduling real-time database transactions[12]. By using EDF, database transactions are classified into two categories, those that have missed their deadlines and those that have not. The latter category can be scheduled using the EDF algorithm, while the former can be kept in background and executed whenever there are no transactions that have not missed their deadlines awaiting services.

A major problem with EDF, when scheduling network traffic, is that all flows receive the same miss rate regardless of their traffic characteristics and deadlines[13]. This makes the standard EDF algorithm unsuitable for situations in which the different flows

have different miss rate requirements since in order to meet all miss rate requirements it is necessary to limit admissions so as to satisfy the flow with the most stringent miss rate requirements.

MATERIALS AND METHODS

The research methodology is based on breaking up the simulation into three stages:

Decomposition: Breaks up the real problem into a set of autonomous elements that interact between them and whose interactions reproduce the real problem.

Modeling: That models each element by defining its functional capabilities, its behaviors and its interaction modes with other elements. This is done through building different java classes for the different elements.

Description: Description of possible actions between the elements by defining the laws which control them.

The performance evaluation technique used for this research is the simulation technique. Simulation is one of the most widely used operation-research and management science techniques. It is defined as the imitation of the operation of a real-world process or system over time. Thus, simulation modelling can be used both as an analysis tool for predicting the effect of changes to existing systems and as a design tool to predict the performance of new systems under varying sets of circumstances.

The specific nature of the developed simulator is considered as a Dynamic Stochastic Discrete event simulation. A dynamic simulation model represent a system as it evolves over time, stochastic is defined as the simulation model contains some probabilistic such as random input components. Finally, discrete event system is one for which the state variables change instantaneously at separated points of time.

System structure of the hierarchical Diff-EDF: The goal of the Hierarchical Diff-EDF scheduling algorithm is to guarantee that a flow's deadline miss rates meet its pre-specified QoS requirements and achieve the high utilization. In EDF scheduler, low priority flows, such as Non-Real-Time traffic, can starve as it is characterized with long lead-times. Despite EDF provides stable QoS guarantees to high priority flows, such as Real-Time traffic, the deadline miss rates of the low priority flows can be unacceptably high. Figure 1 models the EDF scheduler.
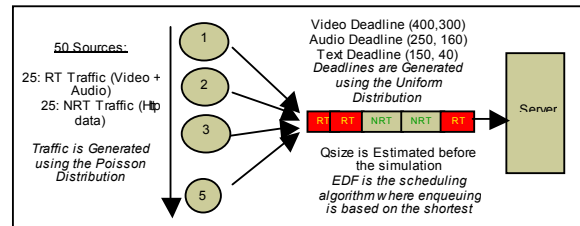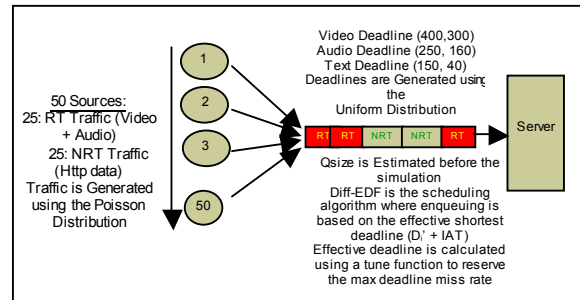

Fig. 1:   EDF scheduler

Fig. 2:   Diff-EDF scheduler

By analyzing the previous figure, three main drawbacks were discovered in using the EDF to schedule real-time packet network traffic:

* All flows receive the same miss rate regardless of their traffic characteristics and deadlines. This makes the standard EDF algorithm unsuitable for situations in which the different flows have different miss rate requirements since in order to meet all miss rate requirements it is necessary to limit admissions so as to satisfy the flow with the most stringent miss rate requirements.

* Packet Starving for the Non-Real-Time traffic. Since Real-Time Traffic is characterized with short lead-times (time until their deadline expires), then it receives high priority comparing to the Non-Real-Time Traffic, which leads to packet starving.

* A random Assignment of Network traffic (Real-Time and Non-Real-Time). As mentioned before, the FCFS scheduling algorithm is widely adopted for best-effort traffic. Having only one service discipline forces all traffic, regardless of their characteristics, to follow the same scheduling algorithm, in our case the EDF.

To overcome the first drawback of the EDF, a new Diff-EDF priority assignment algorithm is proposed[13]. The Diff-EDF scheduling algorithm considers each flow as having stochastic traffic characteristic, a stochastic deadline and a maximum allowable miss rate. Figure 2 shows a representative model for the Diff-EDF scheduler.
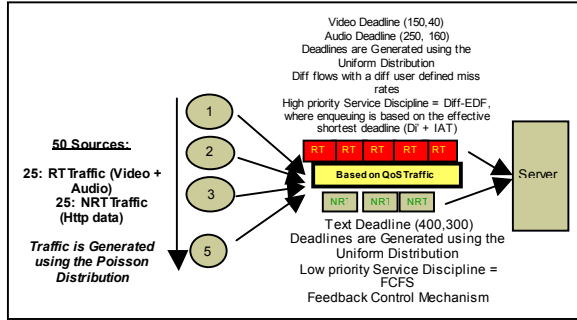
Fig. 3: Hierarchical Diff-EDF scheduler

Again by analyzing the Fig. 2, it is obvious that the last two drawbacks are still discovered when Diff-EDF is used to schedule non-real-time network traffic. As a result, it would be desirable to have a scheme which allows the individual deadline miss rates of different flows to be distinct and controllable. Our proposed Hierarchical Diff-EDF scheme satisfies this objective. It can meet the real-time needs of such applications, by using the Diff-EDF scheduler, while continuing to provide best effort service to non-real time traffic through depending on the strength of the FCFS scheduler. The Hierarchical Diff-EDF features a feedback control mechanism that detects overload conditions and modifies packet priority assignments accordingly. Figure 3 shows a representative model for the Hierarchical Diff-EDF scheduler.

The Hierarchical Diff-EDF scheduler uses a tuning method, or marker, that adjusts the deadlines of the incoming packets by adding a constant to the relative deadlines before the packets are placed into different queues based on the traffic type. Different constants are added to different flows and the modified deadlines are known as "Effective Deadlines".

Packet processing: On receiving each packet from a certain packet flow (assume packet flow j), Diff-EDF performs the following operations:
* Identify the associated flow for the packet using Packet Type as an ID label (we define 1 for http data, 2 for audio and 3 for video) and lookup up the adjustment constant B_j. The relative deadline is then changed to the effective deadline according to the following equation: $D_{e,j} = D_j + B_j$
* Perform the ordinary EDF scheduling using the effective deadline. That is, the packet's absolute deadline is now given by:

$$D'_{e,j} = D_{e,j} + t_a$$

Where t_a is the packet's arrival time and insert the packet into the Diff-EDF queue in the order of increasing absolute deadline (smallest absolute deadline is at queue head thus served first).

Analysis of the Hierarchical Diff-EDF

Assumptions and notations: Assume we have K packet flows and we want to determine whether they can be scheduled so that their QoS requirements are met. Each flow j is characterized by:
* A packet inter-arrival distribution (Exponential Distribution), with a mean of $1/\lambda_j$. Let $\Lambda = \sum_{j=1}^{k} \lambda_j$, the total arrival rate.
* A packet service requirement distribution (Exponential Distribution), with a mean of $1/\mu_j$.
* A soft deadline $D_j > 0$. For each flow j that is randomly drawn from a distribution G_j (in our case the uniform distribution), then D_j represents the mean of G_j. Let $\overline{D}_j = \sum_{j=1}^{k} \lambda_j D_j / \Lambda$, the mean packet deadline averaged over all flows.
* Define $\rho_j = \lambda_j / \mu_j$ the traffic intensity of flow j and $\rho = \sum_{j=1}^{k} \rho_j$. In addition, we define $\alpha_j = \rho_j / \rho$, the faction of the traffic intensity that is attributed to flow j.
* A QoS requirement ϕ_j, interpreted as the requirement that the long run average fraction of flow j's packets missing their deadlines D_j must not exceed ϕ_j.
* The RTQT analysis used in this work models the workload process as a Brownian motion with drift $-\theta$, where:

$$\theta = \frac{2(1-\rho)}{\sum_{j=1}^{k} \lambda_j (\lambda_j^2 + \mu_j^2)/\mu_j^2}$$

Deadline miss rate prediction: Prediction of miss rate, per each flow j, is based on the RTQT analysis of the EDF algorithm. It had been found that when all flows have the same deadline miss rate, then it can be computed by:

$$\phi_j = e^{-\theta \overline{D}}$$

For Hierarchical Diff-EDF, we will adjust the deadlines of each flow j by adding a constant B_j to the deadline. (Constant B_j can be either positive or negative value). Using the above equation, it is obvious that when using Hierarchical Diff-EDF, the deadline miss rate for each flow can be computed by:

$$\phi_j = e^{-\theta(\overline{D'} - B_j)}$$

Where: $\overline{D'} = \sum_{j=1}^{k} \alpha_j (D_j + B_j)$

Determination of B_j's: As we mentioned before, one of the Hierarchical Diff-EDF system components is the Marker. A Marker adjusts the deadlines of incoming packets by adding a constant B_j, different constants are added to different flows, to the relative deadlines before the packets are placed into the queue. After tuning the Diff-EDF system to achieve the best QoS flow's requirements, it had been found that the constant values of B_j could be computed by:

$$B_j = \frac{1}{\theta} \log \frac{\phi_j}{\phi_1}, 1 \le j \le k$$

Where $1/\theta$ is the mean of the exponential stationary distribution of the workload process, ϕ_j is the deadline miss rate of the flow j, ϕ_1 is the smallest deadline miss rate among the flows and k is the number of the flows to be serviced.

If we assume B_1 is the constant to be added for the deadlines in the video flow and that the video flow has the smallest deadline miss rate, then by applying the above equation: $B_1 = 1/\theta \log 1 = 0$. Hence, once the B_j for the high priority flow is determined, the Diff-EDF system will select a much larger values of B_j's for the flows to be run at low priority.

Generating arrival packets: In order to generate the arrival packets, a number of arguments must be determined as the following:
* Number of sources in the system as s.
* Number of flows in the system as k.
* Total Number of Arrival rate for all flows j as $\lambda_T, 1 \le j \le k$.
* Arrival rate per flow j as $\lambda_j = \lambda_T / k$.
* Inter-arrival mean per flow j as $1/\lambda_j$
* Relative Deadline range per flow j as (QoS_{max}, Qos_{min}).

It is worthy to mention here that the total number of arrival rate in the system should equal the summation of all the arrival rates per each flow j as in the following equation: $\lambda_T = \sum_{j=1}^{k} \lambda_j$.

Now, after determine the arguments we start generating the packets. Two Steps were carried out:

1. Using the Exponential Distribution, with a mean of $1/\lambda_j$, to generate the inter-arrival time of the different packets.
2. Using the Uniform Distribution, with a mean of (QoS_{max}, Qos_{min}), to generate the relative deadlines for the different packets.

An important issue that been taken into consideration when generating the arrival packets was to ensure that the Traffic Generator always obtain an initial seeds for the different streams. This mechanism will ensure that each flow is following a specified seed, when its packets are generated, to reflect the real-world traffic and leads for high accuracy. To do that, a Seed Initializer method is used to initialize seeds for the variety streams based on the defined mathematical techniques in[14].

Determination of effective deadlines range: As we mentioned earlier, the Uniform Distribution with a mean of (QoS_{max}, Qos_{min}) is used to generate the relative deadlines for the different packets. QoS_{max} and QoS_{min} are the Effective Deadlines range for each flow *j*. Prasad Calyam and Chang-Gun Lee (Characterizing voice and video traffic behavior over the Internet, 2005) had built a voice and video traffic measurement testbed to determine their effective deadline ranges. The Good range corresponds to delay values of (0-150) ms, the Acceptable range corresponds to delay values of (150-300) ms, while the Poor range corresponds to delay values > 300ms. Now, by observing Fig. 4 and 5, we can conclude:
* The Real Good Range for the video traffic between (40, 150).
* The Real Good Range for the audio traffic between (60, 160).
* The Acceptable Range for both video and audio traffic between (150, 300).
* The poor Range for both video and audio traffic is greater than 300.

Based on the above real results, collected in large-scale Internet, we choose our effective deadline, to achieve the highest system throughput, in the following manner:
* Effective Video Deadline in the Good range (40, 150).
* Effective Audio Deadline in the Acceptable range (250, 160).
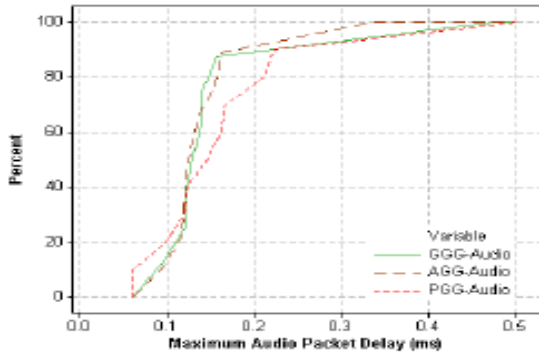* Effective Text Deadline in the Poor range (assume 400, 300).
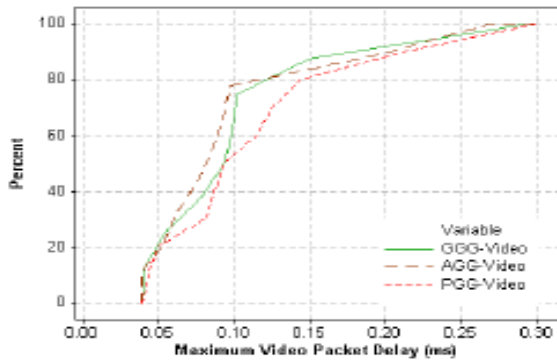
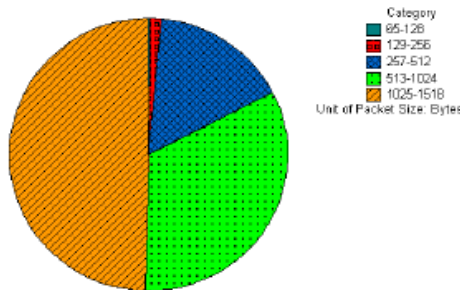Fig. 4:   Maximum audio packet delay

Fig. 5:   Maximum video packet delay

Fig. 6:   Packet distribution

*   The reason to choose the Effective deadline for the text traffic in the poor range is coming from the fact that it is less sensitive to the delay compared with the multimedia traffic.

System queues implementation: In Hierarchical Diff-EDF system, two different queues were implemented as the following:
1.   Diff-EDF queue: This queue is implemented using Sorted Linked List, where the sorting is based on the min value of $(D' + i_{at})$. The reason to choose this type of Linked List is to reduce scheduler complexity, so that rather than the scheduler will

spend the time in picking up the shortest lead time packet to get serve, the queue is ready to be served starting from the queue head. This type of queue will serve all real-time flows (video and audio).
2.   FCFS Queue: This queue is implemented using Linked List. The events are queued based on their Inter-arrival time (i_{at}), with smallest i_{at} at the head of the queue. This type of queue will serve the non-real-time traffic (http data).

Feedback control mechanism: The Hierarchical Diff-EDF scheduling algorithm features a feedback control mechanism that detects overload conditions and modifies packet priority assignment accordingly. To do that, the algorithm is implemented with a feedback control mechanism (Threshold limitation). In other words, the server always serves the packets in the Diff-EDF queue (high priority) and serves the FCFS queue (low priority) if either the Diff-EDF queue is empty or the FCFS queue reaches its threshold value. Now, after tuning the system to achieve the highest performance, through meeting all the flows deadline miss rates ϕ_j, it has been found that the threshold value is when $FCFSQsize >= 0.9\lambda_T / k$. On the other hand, the stopping case was found when the value approaches to $0.7\lambda_T / k$.

System parameters: For this system a number of parameters were set as the following:
*   Packet Size: the packet size was chosen to be of 1500 Byte. The reason to choose this value is that almost more than 50% of the traffic being propagated has a packet size of 1500 Byte (Fig. 6).
*   Bandwidth B_w: the bandwidth was chosen to be 3 Mbps. The reason to choose this value is based on the following facts:
    *   Video traffic consumes a highest bandwidth with a value close to 5 Mbps.
    *   Audio traffic roughly consumes 3 Mbps.
    *   Low traffic consumes about 1 Mbps.
*   The Aggregate Average Bandwidth = (5 Mbps + 3 Mbps + 1 Mbps) / 3 = 3 Mbps
*   Total Arrival Rate λ_T: the simulation was carried out for λ_T start at 5000 packet up to 60000 packet with a 5000 packet simulation step.
*   Mean Service μ: the mean service was calculated with the following equation:
*   Mean Service (μ) = 8 * PacketSize / B_w.
*   Number of Sources s: the simulation was carried out with a 50 generated sources.

* The Experiments were carried out with three different flows; two of them are real-time traffic (video and voice) while the third flow is non real-time traffic (http data or text).

Comparative analysis: The simulation has been run for arrival rates (λ_T) of 10000 – 60000 packets with an increment step of 5000 packets. The bandwidth is assumed to be 3 Mbps while the packet size 1500 Byte. The analysis elaborates different performance metrics with a focus on the miss rate values per each flow j.

Here, four graphs were plotted to compare the performance of the three scheduling algorithms for the different flow j. Figure 7 shows the packet miss rate of the video flow when using each of the three scheduling algorithms. The results show that when the system is moderately loaded the three scheduling algorithms give almost the same results. However, when the system is overloaded it is obvious that the EDF performance degrades rapidly while the Hierarchical Diff-EDF scheduler shows the best packet serving with minimum miss rate.

Figure 8 shows that when the system is overloaded the Diff-EDF scheduler gives the lowest miss ratio compare to both EDF and Diff-EDF schedulers. The figure also shows that the EDF performance continues to degrade proportionally with the number of generated packets, while the Diff-EDF degradation settle at a certain point.

To compare the miss ratio in the case of the text traffic Fig. 9 is used. The results show that the Hierarchical Diff-EDF scheduler shows a remarkable performance by achieving a minimum miss ratio compare to both EDF and Diff-EDF schedulers.
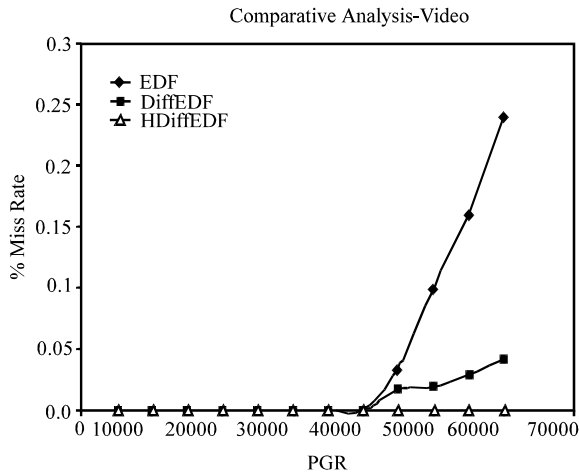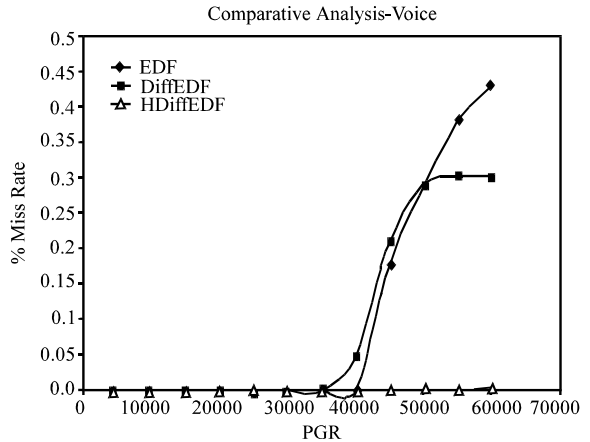

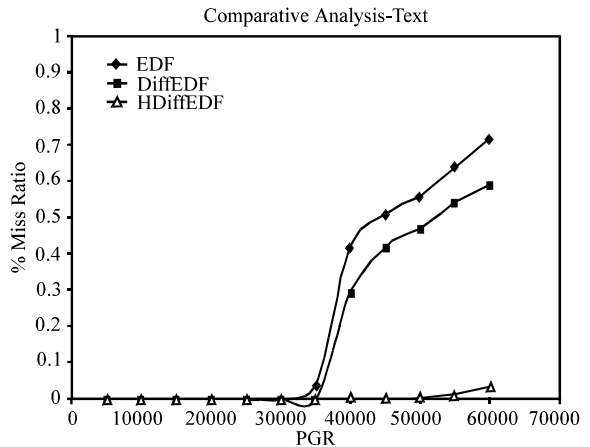Fig. 8: Miss ratio – voice traffic

Fig. 9: Miss ratio – text traffic

Fig. 10: Total miss ratio

Fig. 7: Miss ratio – video traffic
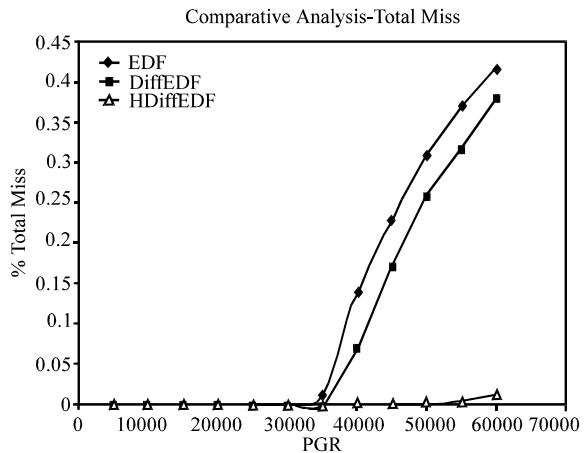
Finally, the total miss ratio for the different flows j is shown in Fig. 10. By analyzing the figure, we can conclude that the Diff-EDF scheduler shows a better

performance of packet serving over heterogeneous network traffic through achieving the minimum miss ratio. This improvement is attributed to the use of the QoS priority based packet serving.

CONCLUSION

In this study, we have presented the new priority assignment scheduling algorithm Hierarchical Diff-EDF (Differentiate Earliest Deadline First), which can meet the real-time needs while continuing to provide best effort service to the non-real time traffic, over heterogeneous real-time network traffic. This Hierarchical Diff-EDF features a feedback control mechanism that detects overload conditions and modifies packet priority assignments accordingly. Also, our scheduler considers each flow as having stochastic traffic characteristic, a stochastic deadline and a maximum allowable miss rate. The Hierarchical Diff-EDF service meets the flow miss rate requirements through the combination of the hierarchal scheduling for the different network traffic classes (real-time and non real-time) and the admission control mechanism that detects the overload conditions to modify packet's priorities. The simulation results show that the Hierarchical Diff-EDF scheduler produces a better performance of packet serving over heterogeneous network traffic through achieving the minimum miss ratio. This improvement is attributed to the use of the QoS priority based packet assignment.

Future research: Proposed future research is as follows:
* To evaluate the performance of the Hierarchical Diff-EDF scheduler under a multiprocessor environment by using the parallel simulation technique.
* To compare the performance of the Hierarchical Diff-EDF scheduler with other different scheduling algorithms such as Round Robin RR, Weighted Fair Queuing (WFQ), Worst-Case Fair Weighted Fair Queuing (WF^{2}Q) and Largest Processing Time First (LPT) under same operating environments.
* To design a more flexible and friendly user interface in a way that allows a user to specify more system parameters with different input data ranges such as bandwidth, packet size, arrival rate (λ), service rate (μ), number of sources, threshold value, type of traffic and size of multimedia stream.
* To test the simulation with other different probability distribution functions rather than using the Poisson distribution and compare the obtained graphs with the collected one.

REFERENCES

1. Srinivasan, A., 2003. Efficient and Flexible Fair Scheduling of Real-time Tasks on Multiprocessors. Chapel Hill.
2. Zhang, H, and D. Ferrari, 1993. Rate-controlled static-priority queuing. In: Proc. of IEEE INFOCOM '93, 1: 227-236.
3. Lu, C.Y., A. Stankovic, G. Tao and H.S. Son, 2002. Feedback control real-time scheduling: framework, modeling and algorithms. Special issue of Real-Time Systems J. on Control-Theoretic Approaches to Real-Time Computing, 23: 85-126.
4. Dertouzos, M., 1974. Control robotics: The procedural control of physical processes. In: Proc. of IFIP Congress, pp: 807-813.
5. Jackson, J.R., 1955. Scheduling a production line to minimize maximum tardiness. Research Report 43, Management Science, University of California, Los Angeles, CA.
6. Towsley, D. and S.S. Panwar, 1991. Optimality of the stochastic earliest deadline policy for the G/M/c queue serving customers with deadlines. COINS Technical Report 91-61, Univ. Massachusetts.
7. Giorgio, C.B. and J. Stankovic, 1993. RED: Robust earliest deadline scheduling. In: Proc. 3rd Intl. Workshop on Responsive Computing Systems, pp: 100-111.
8. Giorgio, B., M. Spuri and F. Sensini, 1995. Values vs. deadline scheduling in overload conditions. IEEE Real-Time Systems Symposium.
9. Firoiu, V., J. Kurose and D. Towsley, 1997. Efficient admission control for EDF schedulers. In: Proc. of IEEE INFOCOM '97, 1: 310.
10. Stankovic, A., M. Spuri, K. Ramamritham and G.C. Buttazzo, 1998. Deadline scheduling for real-time systems – EDF and related algorithms. Kluwer Academic Publishers, Boston, Massachusetts, USA.
11. Tak-Wah, L., T.W. Johnny Ngan and K.K. To, 2004. Performance guarantee for EDF under overload. J. Algorithms, 52: 193-206.
12. Harista, J.R., M. Livny and M.J. Carey, 1991. Earliest deadline scheduling for real-time database systems. IEEE Real-Time Systems Symp.
13. Zhu, H.F., J.P. Lechoczky, J.P. Hansan and R. Rajkumar, 2005. Diff-EDF: A simple mechanism for differentiated EDF service. Proc. of the 11th IEEE Real-Time and Embedded Technology and Applications Symp.
14. Calyam, P. and C.G. Lee, 2005. Characterizing voice and video traffic behavior over the Internet. Intl. Symp0 on Computer and Information Sciences (ISCIS).