International Journal of Soft Computing 2 (4): 520-530, 2007

Hierarchical Diff-EDF: An Agent Based Scheduler for 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 heterogeneous 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 packet networks by the use of proper packet scheduling algorithms. Similar to the trends of computer revolution, many scheduling algorithms have been proposed to meet this goal. The First-Come-First-Served (FCFS), which is mostly used in conventional networks, has been widely adopted for best-effort traffic. In addition, many scheduling algorithms have also been proposed to provide different schemes of QoS guarantees. Among which include the Earliest Deadline First (EDF) and the Differentiated-EDF (Diff-EDF). In this study, we propose a new priority assignment scheduling algorithm, Hierarchical Diff-EDF, which can meet the real-time needs while continuing to provide best effort service over heterogeneous real-time network traffic. The Hierarchical Diff-EDF service meets the flow miss rate requirements through the combination of single step hierarchal scheduling for the different network flows (video, audio and text) and the admission control mechanism that detects the overload conditions to modify packets' priorities. The implementation of this scheduler is based on the multi-agent simulation that takes the inspiration from object-oriented programming. The implementation itself is aimed to the construction of a set of elements which, when fully elaborated, define an agent system specification. When evaluating our proposed scheduler, it was extremely obvious that the Hierarchical Diff-EDF scheduler performs much better than both EDF and Diff-EDF schedulers.

Key words: Hierarchical Diff-EDF, agent based scheduler, heterogeneous, real time packet networks, FCFS

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 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 such 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.

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

Real-time systems: A real-time system is typically composed of several or sequential tasks with timing constraints. In most real time systems, tasks are invocated 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 (Srinivasan, 2003). Thus, the relative deadline parameter is used to specify the timing constraints of the jobs.

A real-time system has two notions of correctness: logical and temporal (Srinivasan, 2003). 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 correctness). However, selecting appropriate methods for scheduling activities is one of the important considerations in the design of a real-time system (Zhang and Farrari, 1994) 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 (Lu *et al.*, 2002). 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.

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 (Dertouzos, 1974). Also, Jackson's rule (Jackson, 1955) 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 (Towsley and Panwar, 1991).

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 (Locke, 1986). 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 Buttazzo and Stankovic (1993). 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, Buttazzo *et al.* (1995) 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 Firiu *et al.* (1997) 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 (Stankovic *et al.*, 1998).

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. Lam *et al.* (2004) 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 (Harista *et al.*, 1991). 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 (Zhu *et al*., 2005). 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.

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.

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 soas 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

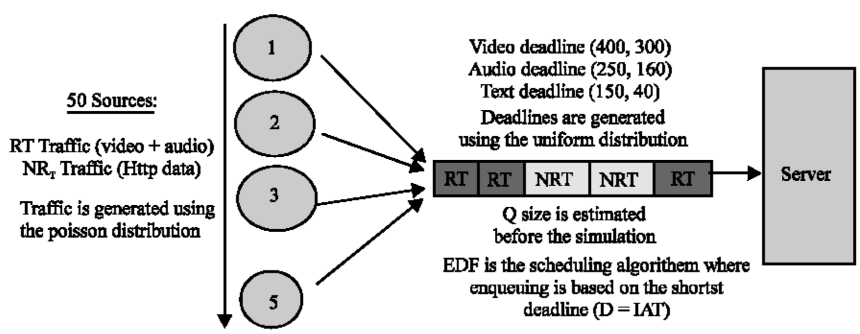

Fig. 1: EDF scheduler

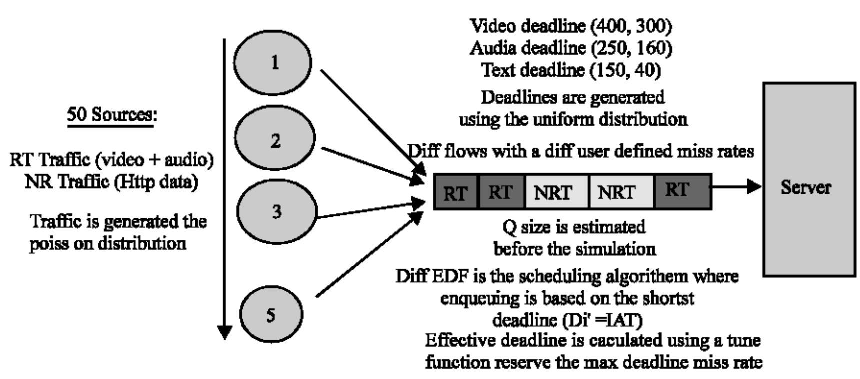

Fig. 2: Diff-EDF scheduler

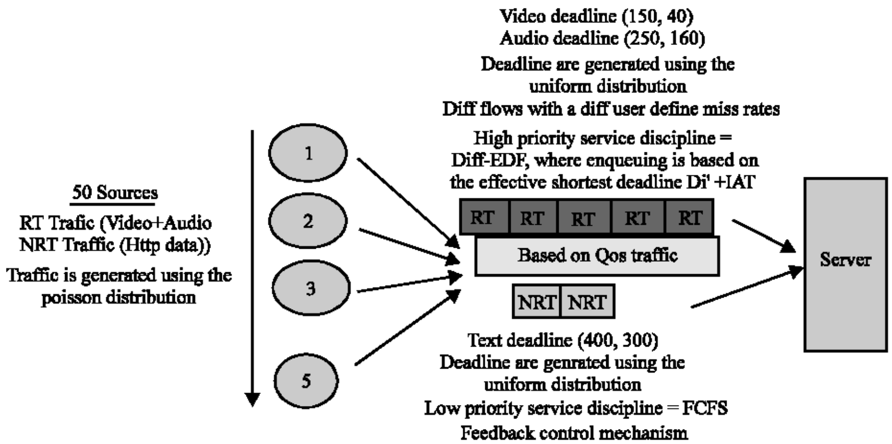

Fig. 3: Hierarchical Diff- EDF scheduler

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 (Zhu *et al*., 2005). 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.

Again by analyzing the above figure, 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

$$D_j = \sum_{j=1}^{k} \lambda_j D_j / \Lambda$$

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 ϕ_j.

- A QoS requirement ϕ_j, interpreted as the requirement that the long run average fraction of flow j's packets missing their deadlines must not exceed.
- The RTQT analysis used in this work models the workload process as a Brownian motion with drift ; 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. The basic methodology can be found in (Lehoczky, 1996, 1998; Zhu *et al.*, 2002). 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:

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

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

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 Bj , 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 can 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, is the deadline miss rate of the flow j, 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 λT, $1 \le j \le k$.
- 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:

- Using the exponential distribution, with a mean of $1/\lambda_j$, to generate the inter-arrival time of the different packets.
- 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 (Marse and Roberts, 1983).

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. Calyam and Lee (2005) have 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

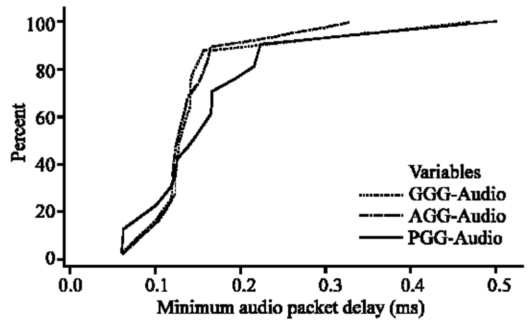

Fig. 4: Minimum audio packet delay

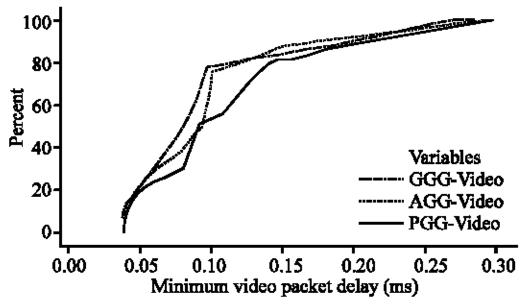

Fig. 5: Minimum video packet delay

Acceptable range corresponds to delay values of (150-300) ms, while the Poor range corresponds to delay values > 300 ms. 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).

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:

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).

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 >=0.9λ/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:

Acket 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 as shown in 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.

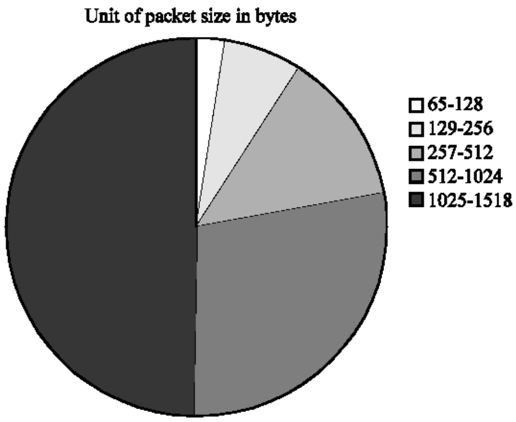

Fig. 6: Packet size distribution

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.
- 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).

AGENT-BASED SIMULATION STAGES

In a discrete event simulation the time and nature of future events is computed in a predetermined fashion from the list of past events which have occurred. Thus the designer of a simulation will typically pre-specify all possible transitions and will not leave the definition of state transitions to entities which are not fully pre-determined. Thus it would be very useful to introduce agents in a simulation whose behavior is determined by specific circumstances and through adaptive behavior. The alternative we propose is that, in addition to the usual attributes of a discrete event simulation (such as event lists and possibly random number generator driven events), a simulation should contain a certain number of agents. These agents store information during a simulation and use it to modify their behavior during that same simulation or during distinct simulation runs.

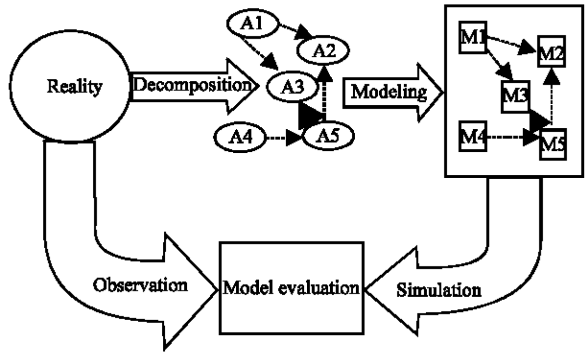

Fig. 7: Agent based simulation stages

From the point of view of agent-based simulation, we break up our simulation into three stages as shown in Fig. 7:

First stage: Includes the decomposition of the real phenomenon into a set of autonomous elements that interacts between each other and whose interactions reproduce the real phenomenon.

Second stage: Includes modeling each element by an agent and definition of its knowledge, its functional capacities, its behaviors and its interaction modes with other agents.

Third stage: Includes the description of possible actions between agents, by defining the environment in which these agents evolve and the rules which control them.

By taking again the three stages described in Fig.7, we place ourselves in an agent context. Thus, we follow the steps of a multi-agents simulation process:

System decomposition: A queuing model is an infrastructure that gathers six types of elements: server, scheduler, queue, source, packet and clock. In addition to the model itself, we have to create a new element (Main) that will be the principle actor of the system behavior.

Modeling each element by an agent: With regard to the elements of the environment, we consider two types of elements: static (passive) agents and dynamic (active) agents. To bring great dynamicity to the system and preserve computer resources we decide to model server, scheduler, queue and main as active agents while source, packet and clock as passive agents.

Description of possible actions between the agents: There are many possible actions between the agents, i.e., messages exchange between the agents to take certain action, change their policy, or update their information.

ARTAS ARCHITECTURE

Our Adaptive Real-Time Agent Scheduling (ARTAS) architecture is a three-layered architecture as depicted in Fig. 8. At the lowest layer, we assume having a Real-Time Operating System (RTOS). Above that are the real-time middleware services which consist of the real-time scheduling agent and the real-time queue agent. At this layer, all tasks are scheduled based on the implemented algorithm, i.e. in our case the Hierarchical Diff-EDF scheduler. Finally, the real-time agent services layer extends the scheduling services resulting in task completion. Moreover, the top layer provides a service for real-time agent communications and interactions.

RT packet agent: Communication among agents in this architecture is performed through a packet request for service from one agent to the others. Each packet request has a formal description of <A, I, D >, where A represents the source to generate this packet request, I is the flow priority to which this packet belongs and D represents the deadline by which the request packet must be completed. If the servicing agent cannot complete this request before its deadline expires then it will be discarded.

RT main agent: As we mentioned earlier, the main agent is the principle actor of the system behavior. Hence, all packet requests should be sent through this agent. It is the agent to control the entire environment through communications with the other agents in the system. The Main agent enforces other agents' policies, disciplines and actions.

RT service agent: This is an active agent which is responsible for packet request completion. It is the main agent which tells this agent when to change their service discipline. The service agent keeps track of the missed packet and reported them to the main agent.

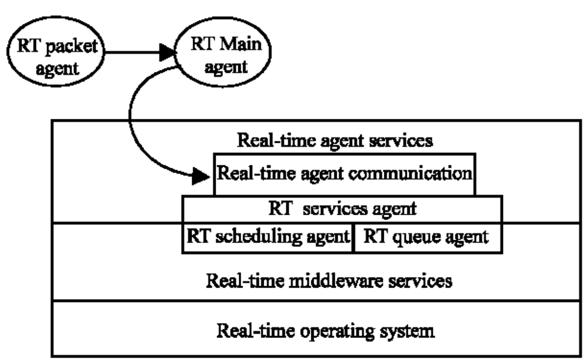

Fig. 8: ARTAS architecture

RT scheduler agent: Our real-time agent scheduling algorithm performs schedulability based on the proposed Hierarchical Diff-EDF algorithm. The packet request is scheduled depending on their original flow, i.e. RT or Non-RT flow. The scheduler agent performs the Marker operation on the incoming packets to achieve the best QoS flow's requirements.

RT queue agent: This is an active agent which treats the incoming packets and place them in the appropriate defined queue based on their flow characteristics requested by the main agent. The queues have limited size and provided with threshold. The queue agent monitors the queue threshold and interacts with the Main agent to inform the queues' status. It also, has the authority to control the queues' filling rate in real-time and the number of discarded packets according to their class.

AGENT MODEL

Our RT Agent model allows for the expression and enforcement of timing constrains on real-time agent interactions. The model is based on the assumption that agents may be able to perform their tasks in variety ways. It is made up of Real-time Agents (RTAgent) and a set of communications among the real-time agents. Figure 9 displays the active elements of the model.

COMPARATIVE ANALYSIS

For evaluating our proposed Hierarchical Diff-EDF scheduler, we present a simulation experiment to study the performance of the Hierarchical Diff-EDF against both EDF and Diff-EDF schedulers. 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.

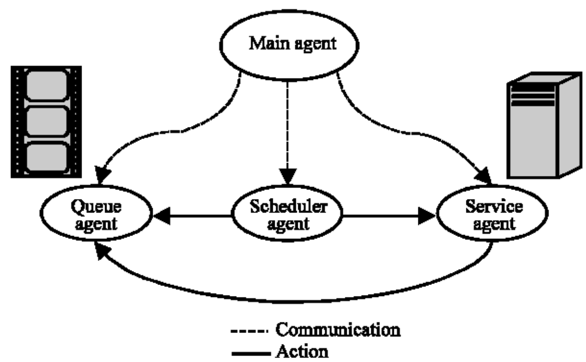

Fig. 9: Agent model

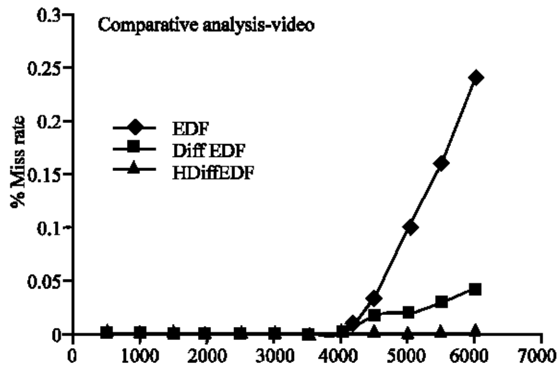

Fig. 10: Miss ratio-video traffic

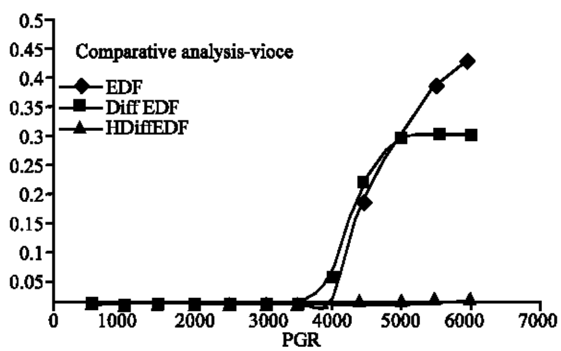

Fig. 11: Miss ratio-voice traffic

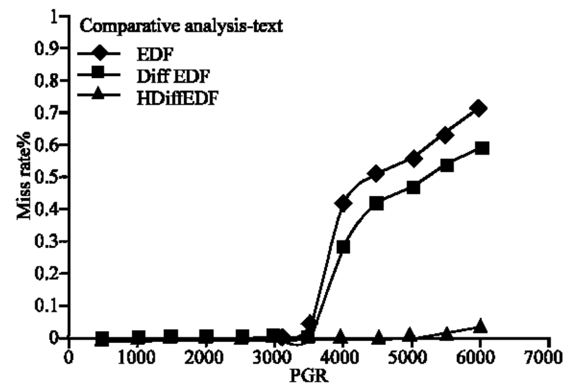

Fig. 12: Miss ratio-tax traffic

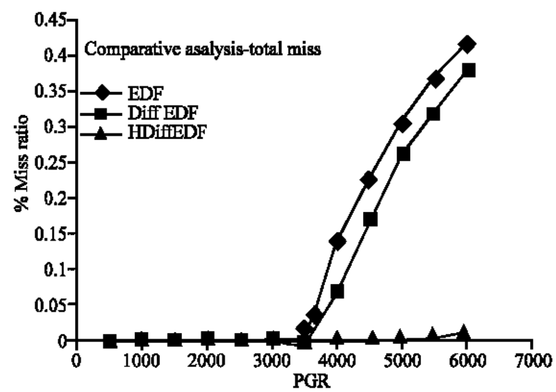

Fig. 13: Total miss ratio

January 1st, 2002 In this study, four graphs were plotted to compare the performance of the three scheduling algorithms for the different flow j. Figure 10 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 11 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 Fig. 11 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. 12 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.

Finally, the total miss ratio for the different flows j is shown in Fig. 13. 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

For the past decade, a significant amount of research in data networks and telecommunications has been devoted to providing different levels of service guarantees. In this paper, we have presented the new priority assignment scheduling algorithm Hierarchical Diff-EDF which met the real-time needs while continuing to provide best effort service to the non-real time traffic, over heterogeneous real-time network traffic. The 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.

During the multi-agent simulation process, we identified our simulation stages by decomposing the system into a number of elements, modeling each element with agent and last describing the possible actions between these agents. Also, we introduced our three-layered ARTAS architecture and agent model. The simulation results showed 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.

Finally, our research study introduced four contributions. First, we showed that considering a single step hierarchical scheduling allowed for a significant enhance in service guarantees for the different network flows. Second, we proved that deploying a feedback control mechanism to allocate service enforced the desired service guarantees. Third, we demonstrated that our proposed scheduling algorithm can achieve high speed performance with minimum miss rate. Last and fourth, we showed that adopting a multi-agent environment during the system implementation results in refining the system design and management which lead for higher performance.

FUTURE RESEARCH

Packet-scheduling disciplines are necessary to satisfy the QoS requirements of delay-sensitive applications and ensure that real-time applications and best effort traffic can coexist on the same network structure. Among packet-scheduling disciplines for providing QoS guarantees to different applications, including real-time services, two classes of algorithms have received particular attention (Fabio and Sivaraman, 1998; Zhang, 1995): Those based on Generalized Processor Sharing (GPS) and those based on Earliest Deadline First (EDF).

As a future work, in situations where the GPS is preferable than EDF, thorough performance evaluation studies between the Hierarchical Diff-EFD and GPS scheduling algorithms can be carried out. Furthermore, the ARTAS architecture and agent model, proposed in this work, can be deployed in such studies to experience its efficiency in refining system's design and management.

REFERENCES

Buttazzo, G., M. Spuri and F. Sensini, 1995. Values vs. Deadline Scheduling in Overload Conditions. Proc. 16th IEEE Real-Time Sys. Symposium, pp: 90-99.

Buttazzo, G.C. and J.A. Stankovic, 1993. RED: Robust Earliest Deadline Scheduling. Proc. 3rd Int. Workshop on Responsive Computing Sys. Aus., pp: 100-111.

Calyam, P. and C. Lee, 2005. Characterizing Voice and Video Traffic Behavior over the Internet. International Symposium on Computer and Information Sciences (ISCIS).

Dertouzos, M., 1974. Control robotics: The procedural control of physical processors. Proc. IFIP. Congress, pp: 807-813.

Fabio, M.C. and V. Sivaraman, 1998. Achieving High Utilization in Guaranteed Services Networks Using Early-Deadline-First Scheduling. The 16th IEEE. Int. Workshop on Quality of Service, pp: 209-217.

Firoiu, V., J. Kurose and D. Towsley, 1997. Efficient Admission Control for EDF Schedulers. Proc. 16th Annu. Joint Conf. IEEE Comput. Commun., 1: 310-317.

Harista, J.R., M. Livny and M.J. Carey, 1991. Earliest Deadline Scheduling for Real-Time Database Systems. The IEEE Real-Time Systems Symposium, pp: 232-242.

Jackson, J., 1955. Scheduling a Production Line to Minimize Tardiness. Research Report, Management Science Research Project in University of California, pp: 43.

Lam, T., T.J. Ngan and K. TO, 2004. Performance Guarantee for EDF under Overload. ACM. J. Algorithms, 52: 193-206.

Lehoczky, J.P., 1996. Real-Time Queuing Theory. In: Proc. 17th IEEE. Real-Time Sys. Symposium, pp: 186-195.

Lehoczky, J.P., 1998. Scheduling Communication Networks Carrying Real-Time Traffic. In: Proc. IEEE. Real-Time Sys. Symposium, pp: 470-479.

Locke, C.D., 1986. Best Effort Decision Making for Real-Time Scheduling. Ph.D. Thesis, Carnegie Mellon University.

Lu, C., J.A. Stankovic, G. Tao and S.H. Son, 2002. Feedback Control Real-time Scheduling: Framework, Modeling and Algorithms. Special issue of Real-Time Sys. J. Control-Theoretic Approaches to Real-Time Comput., 23: 85-126.

Marse, K. and S.D. Roberts, 1983. Implementing a Portable FORTRAN Uniform (0,1) Generator. Simulation. 41: 135-139.

Srinivasan, A., 2003. Efficient and Flexible Fair Scheduling of Real-time Tasks on Multiprocessors. Chapel Hill.

Stankovic, J.A., M. Spuri, K. Ramamritham and G.C. Buttazzo, 1998. Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms. Boston: Kluwer Academic Publishers.

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, University Massachusetts.

Zhang, H. and D. Ferrari, 1994. Rate-Controlled Service Disciplines. J. High Speed Networks, 3: 389-412.

Zhang, H., 1995. Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks. Proc. IEEE., 83: 1374-1396.

Zhu, F., J.P. Lehoczky, J.P. Hansen and R. Rajkumar, 2005. Diff-EDF: A Simple Mechanism for Differentiated EDF Services. Proc. 11th IEEE Real-Time and Embedded Technology and Applications Symposium, pp: 268-277.

Zhu, H.F., J.P. Hansen, J.P. Lehoczky and R. Rajkumar, 2002. Optimal Partitioning for Quantized EDF Scheduling. In: Proc. 23rd IEEE. Real-Time Sys. Symposium, pp: 212-222.