Hierarchical AED Scheduling Algorithm for Real-Time Networks

Saleh Moutaz and Subramaniam Shamala
Department of Communication Technology and Networks, University Putra Malaysia, Malaysia

Abstract: *Earlier studies have observed that in moderately-loaded real-time systems, using an earliest deadline policy to schedule tasks results in the fewest missed deadlines. However, when the real-time system is overloaded an earliest deadline schedule performs worse than most other policies. This is due to the earliest deadline giving the highest priority to the tasks that are close to missing their deadlines, thus delaying other transactions that might still be able to meet their deadline. In this research, an enhanced priority assignment algorithm is presented, called the Adaptive Earliest Deadline (AED), which features a feedback control mechanism that detects overload conditions and modifies packet priority assignments accordingly. Using a detailed simulation model, the performance of AED is compared and analyzed with Earliest Deadline First (EDF). Furthermore, an enhanced AED algorithm called the Hierarchical AED is proposed in a manner in which it obtains a better packet-serving performance by using the concept of priority based on Quality of Service (QoS) of network traffic rather than using a random priority assignment when doing the packet group assignment. Finally, the performance of Hierarchical AED scheduling algorithm is compared with both EDF and the AED scheduling algorithms under the same operating environment.*

Keywords: *Hierarchical, QoS, AED, scheduling, real-time network.*

Received December 2, 2004; accepted July 6, 2005

1. Introduction

Recently, many applications of computer networks rely on the ability of the network to provide Quality of Service (QS) 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 a 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 a bounded delay while it can tolerate 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, these algorithms includes Earliest Deadline First (EDF) and Adaptive Earliest Deadline (AED).

2. Real-Time Systems

A real-time system has two notions of correctness: logical and temporal [11]. In particular, in addition to producing correct outputs (logical correctness), such 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; 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 be completed.

Real-time systems can be broadly classified as hard or soft depending on the criticality of deadlines. 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 invocated repeatedly: each invocation of a task is referred to as a job; and the corresponding time of invocation is referred to as the job's release time or job's deadline [11]. Thus, the relative deadline parameter is used to specify the timing constraints of the jobs.

3. Related Work

There are two basic types of scheduling algorithms: on-line and off-line scheduling [9]. In off-line scheduling, the entire set of jobs to be scheduled including relevant information (in particular, the runtime of each job) is known before a scheduling decision is made. In on-line scheduling, a scheduling decision must be made as soon as one or more jobs are available to be started. There is no information about the arrival of additional jobs in the future and the runtimes of the present jobs may or may not be known.

With scheduling systems, jobs are entered into waiting queues after their creation, and the scheduler then selects jobs from these queues as resources become available. Settings in which jobs are available before the scheduler begins planning their execution schedule are called off-line scheduling. In contrast, settings where the scheduler must make a decision for presently arriving jobs while some other jobs are already running in the system are called on-line scheduling [10]. Since in typical workstation networks jobs are permanently created while other tasks are already running and available CPU capacity changes dynamically due to interfering interactive jobs, traditional off-line scheduling algorithms are hardly applicable [1]. On-line scheduling has recently attracted the attention of several researchers and theoretical results have already indicated the performance of on-line scheduling algorithms in terms of lower and upper bounds [9].

This section presents a review of two real-time network scheduling algorithms, Earliest Deadline First (EDF) and Adaptive Earliest Deadline (AED), with a focus on the AED Scheduling Algorithm as being mainly related to this research work.

- *Earliest Deadline First (EDF):* Is a widely used algorithm for online deadline scheduling. It has been known for long that EDF is optimal for scheduling an under-loaded, 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 [5]. Many real-time systems rely on the EDF scheduling algorithm. This algorithm has been shown to be optimal under many different conditions. In spite of these advantageous properties, EDF has one major negative aspect. That is, when using EDF in a dynamic system, if overloading occurs, tasks may miss deadlines in an unpredictable manner, and in the worst case, the performance of the system can approach zero effective throughputs [2]. This aspect of EDF is a well known fact and applies when using EDF in a dynamic system.

  Also, EDF is widely used in scheduling real-time database transactions. When 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.

  However, EDF works very well unless the workload is very heavy or the real-time system is overloaded. In that case, matters may be improved by introducing some congestion control mechanism.
- *Adaptive Earliest Deadline (AED)*: Earlier studies have observed that in moderately-loaded real-time database systems, using an Earliest Deadline policy to schedule tasks results in fewest missed deadlines. When the real-time system is overloaded, however, an earliest deadline schedule performs worse than most other policies. This is due to earliest deadline giving the highest to transactions that are close to missing their deadlines. The AED, a new priority assignment algorithm, features the feedback control mechanism that detects overload conditions and modifies transaction priority assignment accordingly [3]. The AED priority assignment algorithm modifies the classical Earliest Deadline mapping. In the AED algorithm, transactions executing in the system are collectively divided into two groups, HIT and MISS, as shown in Figure 1.
- *Group Assignment*: Each transaction, upon arrival, is assigned to one of the groups. The assignment is done in the following manner: The newly-arrived transaction is assigned a randomly-chosen integer key, I_T. The transaction is then inserted into a key-ordered list of the transactions currently in the system, and its position in the list, pos_T, is noted. If pos_T is less than or equal to HITcapacity, which is a dynamic control variable of the AED algorithm, the new transaction is assigned to the HIT group; otherwise, it is assigned to the MISS group.
- *Priority Assignment*: After a new transaction is assigned to a group, it is then assigned a priority using the following formula:

$$P_T = \begin{cases} (0,\ D_T,\ I_T) & \text{if Group} = \text{HIT} \\ (1,\ 0,\ I_T) & \text{if Group} = \text{MISS} \end{cases} \quad (1)$$

Utilizing this priority assignment scheme, all transactions in the HIT group have higher priority than transactions in the MISS group. Within the HIT group, the transaction priority ordering is Earliest Deadline. In contrast, the priority ordering in the MISS group is Random Priority since the I_T is randomly selected. The I_T component of the priority serves to break the tie for transactions in the HIT group that may have identical deadlines, thus ensuring a total priority ordering. Transactions retain their initial priority assignments for the entire duration of their residence in the system.

The goal of the AED algorithm is to collect the largest set of transactions that can be completed before their deadlines in the HIT group. It tries to achieve this by controlling the size of the HIT group, using the HITcapacity setting as the control variable. The "hit ratio" of a transaction group is defined to be the fraction of transactions in the group that meet their deadlines. Using this terminology, it would be ideally to have a (steady-state) hit ratio of 1.0 in the HIT group and a hit ratio of 0.0 in the MISS group. Achieving this goal would require absolute accuracy in predicting the right HITcapacity size. Therefore, our aim is to maintain a high hit ratio in the HIT group and low hit ratio in the MISS group. The key to achieving this lies in the HITcapacity computation.

- *HIT Capacity Computation*: A feedback process that employs system output measurements is used to set the HITcapacity control variable. The measurements used are HitRatio (HIT) and HitRatio (ALL). HitRatio (HIT) is the fraction of transactions in the HIT group that are making their deadline, while HitRatio (ALL) is the corresponding measurement over all transactions in the system. Using these measurements, and denoting the number of transactions currently in the system by NumTrans, the HITcapacity is set with the following two-step computation:

1. HITcapacity = HitRatio (HIT) * HITcapacity * 1.05.
2. If (HitRatio (ALL) < 0.95) then HITcapacity = Min (HITcapacity, HitRatio (ALL) * NumTrans * 1.25).

The first step of the HITcapacity computation incorporates the feedback process in the setting of this control variable. By conditioning the new HITcapacity setting based on the observed hit ratio in the HIT group, the size of the HIT group is adaptively changed to achieve a 1.0 hit ratio. However, the main goal is not just to have a HitRatio (HIT) of 1.0, but also to achieve this goal with the largest possible transaction population in the HIT group. It is for this reason that step 1 includes a 5 percent expansion factor. This expansion factor ensures that the HITcapacity is steadily increased until the number of transactions in the HIT group is large enough to generate a HitRatio (HIT) of 0.95. At this point, the transaction population size in the HIT group is close to the required number, and the HITcapacity remains stabilized at this setting (since 0.95 * 1.05 = 1.0).

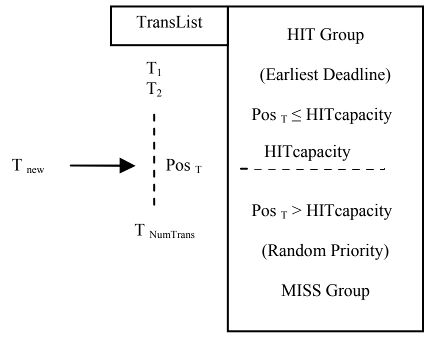

Figure 1. AED priority mapping.

The second step of the HITcapacity computation is necessary to take care of the following special scenario: If the system experiences a long period where HitRatio (ALL) is close to 1.0 due to the system being lightly loaded, it follows that HitRatio (HIT) will be virtually 1.0 over this extended period. In this situation, the HITcapacity can become very large due to the 5 percent expansion factor, that is, there is a "runway" effect. If the transaction arrival rate now increases so that the system becomes overloaded (signaled by HitRatio (ALL) falling below 0.95), incrementally bringing the HITcapacity down from its artificially high value to the right level could take a considerable amount of time (with the feedback process of Step 1). This means that the system may enter the unstable high-miss region of Earliest Deadline as every new transaction will be assigned to the HIT group due to the high HITcapacity setting. To prevent this from occurring, an upper bound on the HITcapacity value is used in STEP 2 to deal with the transition from lightly-loaded condition to an overloaded condition. The upper bound is set to be 25 percent greater than an estimate of the right HITcapacity value, which is derived by computing the number of transactions that are currently making their deadlines.

4. Modeling and Simulation

In this research, three different scheduling algorithms were modeled and analyzed. These scheduling algorithms are: EDF, AED and the new proposed hierarchal AED. In this section, each of the three algorithms will be discussed in details.

4.1. EDF

EDF is widely used in scheduling real-time database transactions. 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.

However, EDF works very well unless the workload is very heavy or the real-time system is overloaded. In that case, matters may be improved by introducing some congestion control mechanism. Figure 2 shows a graphical representation for the EDF scheduling algorithm.

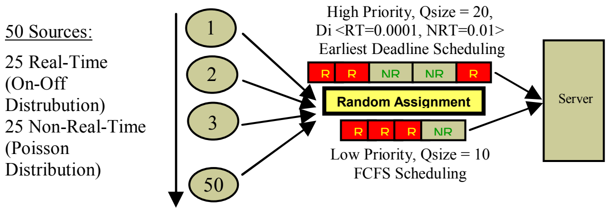

Figure 2. EDF scheduling algorithm.

When modeling this scheduling algorithm, the following assumptions are made:

- *Traffic Type*: Two types of network traffic are generated: Non real-time network traffic (i. e., Text traffic) and real-time network traffic (i. e., Multimedia traffic).
- *System Parameters*: When developing this system, the following parameters have been defined:
  - *Source Management*: This system is considered as a multiple-source system. It uses 50 sources to generate the network traffic. However, since there are two types of network traffic, the sources are divided equally to generate the traffic. It is assumed that sources labeled from 1 to 25 are generating real-time traffic, while sources labeled from 26 to 50 are generating non real-time network traffic.
  - *Estimation of Qsize*: As shown in Figure 2, this system has two queues. The first queue Q1 is considered of higher priority while the second one Q2 is considered of lower priority. This concept means that packets assigned to Q1 have higher priority to be served than packets assigned to Q2. To set both Q1 and Q2 sizes, a Qsize estimation function is used.
  - *Service Discipline*: The packet scheduling algorithm used for Q1 is Earliest Deadline scheduling, while Q2 uses the First-Come-First-Served (FCFS) as a scheduling algorithm. The simulation is run without any feedback control mechanism (no threshold limitation). This means that the server always serves the packets in the high priority queue (Q1), and serve the low priority queue (Q2) if and only if the high priority queue is empty.
  - *InterArrival Times Generation*: To generate the packet Interarrival times, the Poisson distribution is used. However, in the case of generating real-time traffic each source generate a stream of 25 packets at each session. The value of 25 is used since it is considered to be the minimum value of packet stream that allow a normal user to distinct a multimedia traffic.
  - *Packet Assignment*: Once the Interarrival times are generated using Poisson distribution, it must be now assigned to one of the queues. To do that, the Bernoulli distribution is used.

Table 1. System parameters–experimental models.

Parameter	Value
Bandwidth	100 Mbps
Packet Size	250 Byte
Starting Arrival Rate (λ Start)	10000 packet per second
Finishing Arrival Rate (λ End)	60000 packet per second
Interarrival Step	5000 packet
Service Rate (μ)	$2 * 10^{-6}$
Number of Sources	50
Multimedia Stream	25 packet per second

4.2. AED

By analyzing Figure 2, two main drawbacks were discovered in using the EDF to schedule real-time network traffic:

- Under heavy loads ($\lambda > \mu$), no feedback control mechanism is used.
- A random assignment of network traffic (no QoS guarantee). Packet assignment is based on Bernoulli random generator.

Earlier studies have observed that in moderately-loaded real-time database systems, using an Earliest Deadline policy to schedule tasks results in fewest missed deadlines. When the real-time system is overloaded, however, an Earliest Deadline schedule performs worse than most other policies. This is due to Earliest Deadline giving the highest to transactions that are close to missing their deadlines. To overcome the first drawback the Adaptive Earliest Deadline (AED), a new priority assignment algorithm, is used. The AED scheduling algorithm features the feedback control mechanism that detects overload conditions and modifies transaction priority assignment accordingly. Figure 3 shows a graphical representation of the AED scheduling algorithm.

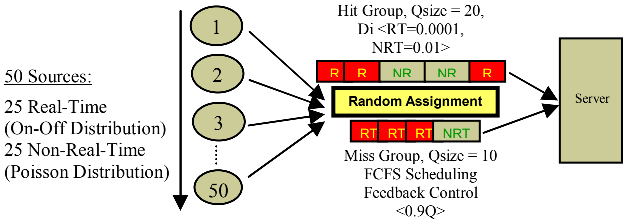

Figure 3. AED scheduling algorithm.

The model development of this scheduling algorithm is the same as the EDF. However, following are the differences:

- *Queue Management*: In this scheduling algorithm, two types of queue are defined: the high priority queue, denoted as the HIT group, and the low priority queue denoted as the MISS group. The details have been elaborated in section 3.
- *Feedback Control Mechanism (Threshold)*: The AED 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 HIT group, and serves the MISS group if the HIT group queue is empty or the MISS group queue reaches its threshold value. The experiments show that the optimal threshold, the point at which the system obtains the lowest packet loss ratio, is taken when the threshold value is 0.9 of the Low priority queue size (MISS group queue).

4.3. Hierarchical AED

Again by analyzing Figure 3, it is obvious that a major drawback is discovered when AED is used to schedule real-time network traffic which is the random assignment of network traffic (no QoS guarantee). Packet assignment is based on Bernoulli random generator.

The AED scheduling algorithm overcomes this drawback of the EDF algorithm by including a feedback control mechanism that detects overload conditions and modifies packet priority assignment accordingly. However, the AED still suffer a drawback in the packet assignment.

Since real-time traffic requires bounded delay while it can tolerates some packet losses, the AED scheduling algorithm can be enhanced by adopting a new priority packet assignment. The hierarchical AED scheduling algorithm enhances the AED algorithm in such a way that it obtains a better packet-serving performance by using the concept of priority based on QoS of network traffic as opposed to using a random priority assignment in the process of packet group assignment. Figure 4 shows a graphical representation for the hierarchal AED scheduling algorithm.

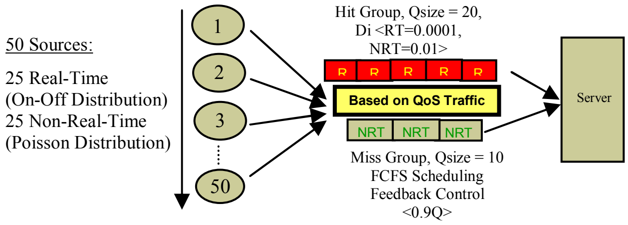

Figure 4. Hierarchal AED scheduling algorithm.

The development of this scheduling algorithm is the same as in the AED, with the only difference in the packet assignment. While AED uses random packet assignment based on the Bernoulli distribution, the enhanced AED scheduling algorithm uses a packet assignment based on QoS traffic. Assignment based on QoS traffic means that if the arrived packet is a real-time one (i. e., multimedia traffic) then it will be assigned to the HIT group (high priority queue). On the other hand, if the arrived packet is a non real-time one then it will be assigned to the MISS group (low priority queue). Doing the packet group assignment in such a way improves the AED scheduling algorithm when this algorithm is used to schedule real-time network traffic. The idea can be proved since real-time traffic requires bounded delay while it can tolerate some packet losses.

5. Comparative Analysis

The main goal of this research is to enhance the performance of the AED algorithm for scheduling real-time network traffic, by using the concept of QoS based packet assignment as opposed to using random packet assignment, and to compare the performance of the hierarchical AED scheduling algorithm with both EDF and the AED scheduling algorithms.

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 100 Mbps while the packet size 250 Byte. The simulation is terminated when the number of departed packets (packets that obtain the service) is equal to the arrival rate (λ_T) at each simulation step. The analysis elaborates four performance metrics: average delay, average buffer (average Qlength), packet loss ratio, and server utilization.

In this section, four graphs were plotted to compare the performance of the three scheduling algorithms. Figure 5 shows the average delay of the system when using each of the three scheduling algorithms. The results show that when the system is moderately loaded all of the three scheduling algorithms give almost the same average delay with preferability to the EDF. However, when the system is overloaded, both the

Hierarchical AED and AED algorithms give the lowest average delay compare to the EDF.

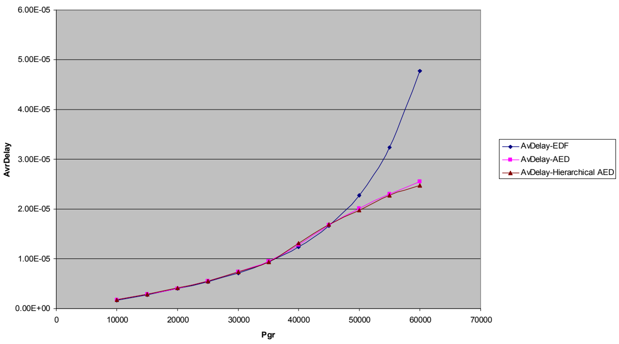

Figure 5. Average delay.

Figure 6 shows the server utilization. The results show that the server is utilized higher when using the hierarchical AED algorithm. This is due to the fact that the hierarchical AED is based on two strengths: the advantage of the AED algorithm in managing the service of the queues and the power of the packet assignment technique which based on QoS priority.

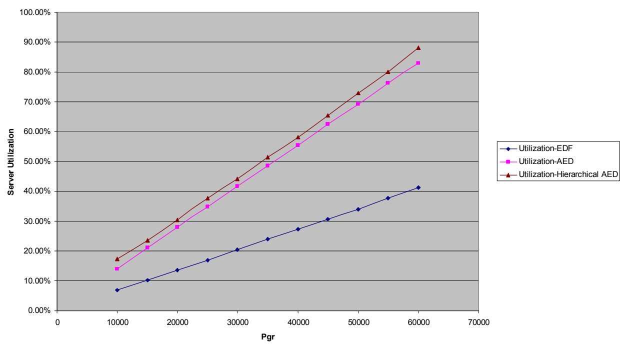

Figure 6. Server utilization.

To compare the packet loss ratio, Figures 7 and 8 are used. Figure 7 shows the packet loss ratio of the Real-Time traffic generated by sources (1-25), while Figure 8 shows the packet loss ratio of the Non Real-Time traffic generated by sources (26-50). In both figures, the Hierarchical AED shows a much better packet loss ratio comparing with both AED and EDF. This improvement is attributed to the use of the QoS priority based packet assignment.

6. Conclusions

In this research, the performance of the Adaptive Earliest Deadline algorithm has been enhanced when scheduling real-time network traffic, by using the concept of QoS based packet assignment rather than using random packet assignment. The performance of the hierarchical AED scheduling algorithm has been compared with both EDF and the AED scheduling algorithms. The results show that when the system is moderately loaded all of the three scheduling algorithms give almost the same average delay with preferability to the EDF. However, when the system is overloaded, both the Hierarchical AED and AED algorithms give the lowest average delay compare the EDF. For the Server Utilization, the analysis shows that the server is more utilized when using the Hierarchical AED algorithm. When comparing the Packet Loss Ratio, the Figures show that the Hierarchical AED gives the lowest ratio comparing with both AED and EDF. These great results are attributed to the use of the QoS priority based packet assignment.

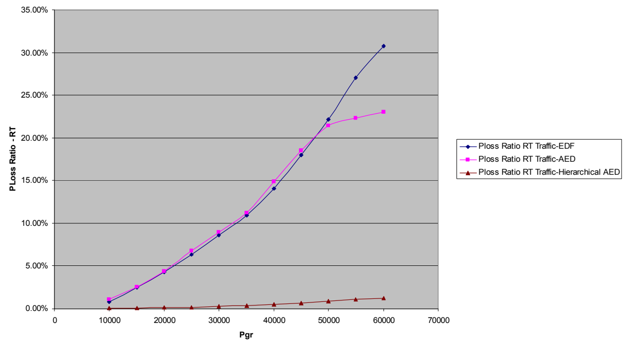

Figure 7. Packet loss of RT traffic.

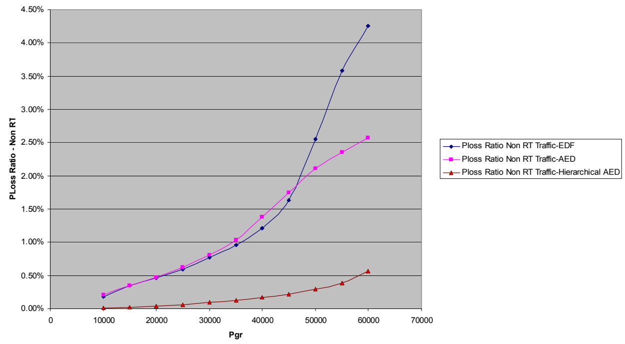

Figure 8. Packet loss of non-RT traffic.

7. Future Research

Proposed future research is as follows:

- To evaluate the performance of the hierarchical AED algorithm, in scheduling different network traffic, under a multiprocessor environment by using the parallel simulation technique.
- To compare the performance of the hierarchical AED algorithm with other different scheduling algorithms such as Round Robin RR, Weighted Fair Queuing (WFQ), Worst-Case Fair Weighted Fair Queuing (WF^2Q) 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] Arndt O., Freisleben B., Kielmann T., and Thilo F., "A Comparative Study of Online Scheduling Algorithms for Networks of Workstations," *Cluster Computing*, vol. 3, no. 2, 2000.

[2] Giorgio C. Buttazzo, and Stankovic J. A., "RED: Robust Earliest Deadline Scheduling," *in Proceedings of the 3rd International Workshop on Responsive Computing Systems*, Lincoln, NH, pp. 100-111, September 1993.

[3] Harista J. R., Livny M., and Carey M. J., "Earliest Deadline Scheduling for Real-Time Database Systems," *in Proceedings of IEEE Real-Time Systems Symposium*, Texas, pp. 232-242, 1991.

[4] Hei X. and Tsang D., "The Earliest Deadline First Scheduling with ActiveBuffer Management for Real-Time Traffic in the Internet," *in Proceedings of the 1st International Conference on Networking-Part 1*, Springer-Verlag, pp. 45-54, 2001.

[5] Lam T. W., Ngan T. W., and To K. K., "Performance Guarantee for EDF Under Overload," *Journal of Algorithms*, vol. 52, no. 2, pp. 193-206, August 2004.

[6] Loher U., "Efficiency of First-Come-First-Served Algorithms," *Information Theory*, vol. 16, no. 21, pp. 108, August 1998.

[7] Marosits T. , and Molnar S., "Traffic Control Methods for High Speed Packet Switched Networks," *in Proceedings of the Polish-Czech-Hungarian Workshop on Circuit Theory, Signal Processing, and Their Applications*, Warsaw, Poland, pp. 12-15, September 2002.

[8] Ng T. S., Stephens D. C., Stoica I., and Zhang H., "Supporting Best-Effort with Fair Service Curve," *in Proceedings of the IEEE Global Telecommunications Conference*, vol. 3, pp. 1799-1807, 1999.

[9] Sagall, "Online Scheduling," *A Survey*, LNCS 1442, Springer, pp. 196-231, 1996.

[10] Shrnoys D. B., Wein J., and Williarnsom D. P., "Scheduling Parallel Machines On-Line," *SIAM Journal on Computing*, vol. 24, no. 6, pp.1313-1331, 1995.

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

[12] Zhao W. and Stankovic J. A., "Performance Analysis of FCFS and Improved FCFS Scheduling Algorithms for Dynamic Real-time Computer Systems," *in Proceedings of the 10th IEEE Real-Time Systems Symposium*, pp. 156-65, December 1989.

Saleh Moutaz received his bachelor degree in computer engineering from Yarmouk University, Jordan in 2000, his master degree in distributed computing from University Putra Malaysia, Malaysia in 2004, and now he is doing a PhD in computer science at University Kebangsaan Malaysia, Malaysia. Currently, he is a lecturer at the Department of Computer Science & Engineering, Qatar University, Qatar. He holds Cisco Certified Network Associate (CCNA), and Cisco Certified Network Professional (CCNP). He worked before in Networking Department at the Ministry of Justice, Saudi Arabia. He also was the network engineer at UBM, Jordan. His research interests include real-time networks, and multi agent systems.

Subramaniam Shamala completed her BSc, MSc, and PhD from University Putra Malaysia in 2002. Currently, she is a lecturer at the Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, University Putra Malaysia. Her research interest includes scheduling algorithms, congestion control, real-time systems, modeling, and simulation.