Comparing FCFS & EDF Scheduling Algorithms for Real-Time Packet Switching Networks

Maen Saleh, *Student Member, IEEE*, and Liang Dong, *Senior Member, IEEE*

***Abstract*— Nowadays, most of the computerized network applications are depending on the ability of the servicing packet switching network to integrate different types of traffic from two main categories, real-time and best-effort classes, and provide different QoS guarantees based on the traffic type. Different methods were applied to provide such guarantees, one of the most efficient method used in packet switching networks is the use of the proper scheduling algorithms. In this paper we have studied two main schedulers, The First-Come-First-Served (FCFS) scheduler which is commonly used in best-effort networks where no QoS guarantees are required, and the Earliest Deadline First (EDF) scheduler which is mostly used to provide different QoS requirements for real-time traffic while continuing to provide a service for best effort data traffic. Three main QoS guarantees had been studied (miss ratio, delay, and average size of the buffer) by using these two scheduler on a set of data packets from two main categories (real-time and best-effort). The study shows the efficiency of using the EDF scheduler in a hybrid network to provide such QoS guarantees among the FCFS scheduler which is more efficient for serving best-effort data traffic.**

I. Introduction

THE early computerized network applications were not concerned or sensitive to any of the QoS guarantees such as the traffic delays or the data losses, so the first wide world network (internet) which had been created in the beginning of 1970s was mainly created to provide services for those types of data traffic, such services are called best-effort services.

Nowadays the internet provides different types of guarantees to the different classes of applications using the it, these guarantees are called the quality of service guarantees (QoS) [1].

The quality of service guarantees can be in different forms, such as bandwidth, miss ratio in terms of packet loss, total packet delays and buffer size; it also could be in a form of any combination of the previous metrics [2]. The applications using this type of network, which provides QoS guarantees on the data traffic are different in there types, some of them are real-time applications in which a stringent QoS guarantees are required from the network, while other applications need no guarantees (best-effort traffic), then the network must be capable of providing such guarantees to the different classes of traffic using this network.

Different methods were applied by packet switching networks to provide the proper QoS guarantees to the traffic uses it, one of the most efficient methods is by using one of the suitable network packet schedulers, where the main functionality is to select a packet from a multiple of received packets to be sent through the network, so the scheduler answers the question: whose turn is now?.

Two main schedulers were studied in this paper, one of them is the First-Come-First-Served (FCFS) scheduler which is commonly used in best-effort networks where no QoS guarantees are required, and the other one is from the priority scheduler category called the Earliest Deadline First (EDF) scheduler which is mostly used to provide different QoS requirements for real-time traffic while continuing to provide a service for best effort data traffic.

II. Network Scheduling

A. FCFS scheduling algorithm

The First-Come-First-Served (FCFS) scheduler is one of the most common schedulers designed for best-effort networks where no QoS guarantees are required. The design and implementation of this scheduling algorithm is very simple without any complications, since the only data required by scheduler to make a scheduling decision is the arrival-time of the packet to the scheduler, then the algorithm take place by serving the packet arrived first without looking to the type of the packet, the packets are considered as equally-likely (same treatment).

The standard FCFS scheduler has no complications in it's design, and it will work efficiently in a best-effort environment, but when we apply it to real-time systems where the time constraints are important and must be guaranteed, the scheduler may suffer of high miss ratio for real-time traffic, since the real-time traffic will be treated in the same manner to the best-effort traffic, so new versions of FCFS scheduler had been created to make it more efficient in real-time systems, one of them depends on changing the location of the new arrived packet to the scheduler. In this algorithm the scheduler has a limited buffer size in which successful scheduled packets will be queued in, once a new packet arrives the scheduler it will be resides in the tail of the buffer (queue) if it's time constraints can be achieved, if we were dealing with the standard FCFS scheduler, then the scheduler will drop the packet if its time-constraints are not met, but using this new version of FCFS there will be another chance to accept the packet by replacing it with the

This work was supported in part by funds from the Faculty Research and Creative Activities Award and the Research Development Award of Western Michigan University.

M. Saleh is with the Department of Electrical & Computer Engineering, Western Michigan University, Kalamazoo, MI 49009 USA (phone: 269-873-0206; e-mail: maen.s.saleh@wmich.edu).

L. Dong is with the Department of Electrical & Computer Engineering, Western Michigan University, Kalamazoo, MI 49009 USA (phone: 269-276-3160; e-mail: liang.dong@wmich.edu).

last packet being in the scheduler before the new packet arrive, if this replacement process didn't conflict any timing constraints for both packets, then the replacement process take place, otherwise the new packet will be dropped [3].

The FCFS scheduler will be inefficient when we are dealing with dynamic real-time systems especially when we are talking about the hard real-time systems, it could be used with real-time systems where there is flexibility in the timing-constraints for the real-time jobs, and it will be more efficient when dealing with best-effort traffic. As we said before the new requirement of the packet switching network is to provide QoS guarantees for the real-time traffic while continuing to serve best-effort traffic, so new scheduling algorithms were built based on this requirement by combining more than one algorithm together in a new algorithm, for example the FCFS had been used in real-time system to serve the best-effort traffic, while another scheduling algorithm more efficient for scheduling the real-time traffic such as Earliest Deadline First (EDF) scheduler had been used to take care of real-time traffic, this implementation needs from the network the ability to distinguish between the different types of traffic, and separate them in different queues then apply the appropriate scheduling algorithm for each type of traffic, by this way the FCFS usage will be more efficient instead of using it alone to deal with both types of traffic [4]. FCFS can be used in real-time systems where the number of real-time tasks are small (non heavy traffic), in this way the advantage of using a simple algorithm will help in decreasing the overhead time due to the scheduler processing and so more efficiency.

B. *Earliest Deadline First (EDF) scheduling algorithm*

This algorithm is one of the priority based scheduling algorithms, which is more efficient in serving real-time traffic. In this priority based algorithm, each real-time traffic will be given a specific priority based on the remaining time for the packet to expire, that is the packet with its deadline is closer to expire will be given a higher priority, so it will be served first, this algorithm is suitable when dealing with real-time systems with packets have different deadlines [5]. As we can see from its functionality, the EDF is very efficient in guaranteeing the QoS requirements for real-time traffic, but studies and researches discovers different disadvantages of the EDF scheduler, the effect of traffic overloading in dynamic environment will make the EDF scheduler inefficient in serving the real-time data packets, and a high miss ratio will be achieved, to solve the problem caused by overloading, a new and modified EDF scheduler was created called the Adaptive Earliest Deadline (AED) scheduler, this scheduler have a monitoring policy over the entire real-time system , this system will observe any overloading happens in the system periodically and send a feedback signal to the scheduler to make a reaction, that is in the case of overloading the scheduler will run a new priority assignment algorithm that guarantees the QoS requirements for the different flows, and then the miss ratio will not be increased, it will stay in its limits [6]. Another disadvantage is the starvation problem, where the best-effort traffic will not be served if there is any real-time traffic, this problem can be solved easily by integrated a different scheduling algorithm for serving the non real-time traffic based on some conditions in the system such as if we use a point of switching from serving real-time to non real-time traffic, this point mainly depends o the number of non real-time traffic arrived to the system and have not been served, a threshold function could be used here as a switching point, so by that we will guarantee the QoS requirements of real-time systems, and still provide services for best-effort traffic. Another disadvantage of using such scheduler is that when we have different flows and each one of these flows has a different miss ratio requirement, the EDF give a fixed miss ratio for all data flows without looking to the specifications to the real-time packets, so the QoS guarantees will not be met for some real-time flows [7].

Hierarchical systems have been studied in mono-processor applications for real-time systems. one of the hierarchal schedulers implemented is a scheduler that consist of two main levels, external (global), and internal (local) levels, the external scheduler uses fixed priority algorithm for its real-time tasks, and the internal one is a priority scheduler also, but it was EDF scheduler. Another implemented scheduler also has two levels, but without using the fixed priority one, in both levels the scheduling algorithm used is the EDF. By applying these two schedulers to a mono-processor and according to the obtained results, the second implementation of the scheduler was better than the first one that used the fixed priority schemes, the results showed that the response time for the processor in the second case is less than it in the first implementation and thus higher utilizations to the mono-processor [8].

III. System Design

A. *FCFS design process*

The system consists of four main components, Data Packets Generator, Arrival-Time generator, Control-Unit, and QoS Measurements Component. In order to measure the QoS requirements of the system, we have chosen a specific traffic sequence of RT and NRT packets with a total of 50 packets, and we fixed the deadline for real-time data packets so that the comparison process will be meaningful.

The system begins by asking the user to insert specific values for the processing time of the scheduler, RT packets deadline, and the period between data packets (uniform distribution), these inserted data will be passed to the data packet generator component and to the arrival-time component, then the data packet generator will generate the sequence of the RT and the NRT packets with the associated deadline for the RT traffic, while the arrival-time component will assign an arrival-time for each data-packet in a uniform distribution.

The packets are then sent to the control-unit based on their arrival-time, the scheduler will be in a waiting state until the packets arrived, when it receives a packet it begins its functionality of scheduling by checking its type (RT or NRT packet), if it was a NRT packet, it just serve it without any conditions, while if it was a RT packet then it checks the time that the packet waits in the scheduler before being in the head of the scheduler queue ready for serving or dropping, if the packet waited for a time that is less than its deadline then it will be served, otherwise it will be dropped. The scheduler continues this process taking into consideration the FCFS property. The QoS measurements component will be working side-by-side with the scheduler, calculating all the QoS metrics and display them.

B. EDF design process

The system in its construction is similar to the FCFS scheduler, but the scheduling algorithm performed by the control-unit (scheduler) will be different.

the scheduler will be in a waiting state until the packets arrived, when it receives a packet it begins its functionality of scheduling by checking its type (RT or NRT packet), if it was a NRT packet, it just serve it without any conditions, while if it was a RT packet then it checks the time that the packet waits in the scheduler before being in the head of the scheduler queue ready for serving or dropping, if the packet waited for a time that is less than its deadline then it will be served, otherwise it will be dropped.

After serving the first arrived packet, it returns back to the initial state but here it may find more than one packet received., in this case it checks the type of the packets, if all of them were non real-time packets, then it will serve the first arrived packet among them (FCFS scheduling), but if there were real-time and non real-time packets then it deals with the real-time packets only, it will check among these RT packets which of them is closer to be expire, this will be done by performing the following equations:

$$Q = C - A$$

Q: Queuing Time, C: Current-Time A: Arrival-Time, then

$$R = Q - D$$

R: Remaining Time, D: Deadline

Then the scheduler will check the packet with the smallest positive remaining time to serve. QoS measurements component will be working side-by-side with the scheduler, calculating all the QoS metrics and display them.

.

IV. Simulation Results

A. FCFS scheduling algorithm

There are many factors affect the QoS requirements of this system, the first factor that affects the QoS requirements is the processing time of the scheduler, so by fixing the RT deadline and the time intervals between the data packets, the system was executed for different values of the processing time, in this scenario we have fixed the RT deadline to be15 ms, and the time periods to be 10 ms.

The first data to be analyzed is the effect of the processing time on miss ratio, as shown from Fig.1, the miss ratio will remain very small (almost zero) for values of processing time below arrival time.

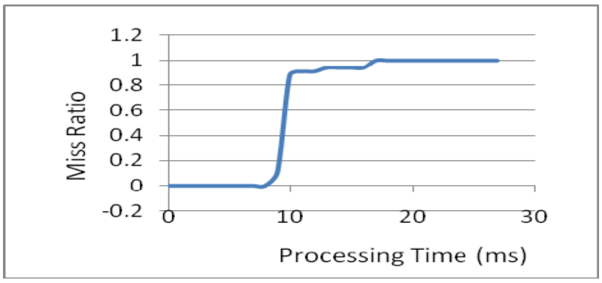

Fig.1. The effect of processing-time on miss ratio

A high transition (down to up) will occur (near 100%) at the period where both processing time and arrival-time are equal, then the changes will be small until it reaches 100% miss ratio.

The second data to be analyzed is the effect of the processing time on total packets delays, as shown from fig.2 the changes of the total delay remain relatively very small for values of processing time below arrival Time, an approximate linear relationship with a large positive slope is then appears after that value of the processing-time (equal to arrival-time).

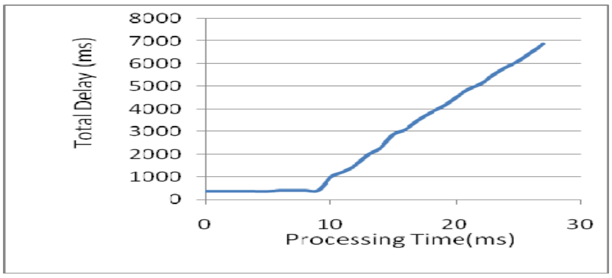

Fig.2. The effect of processing-time on total-delay

The third data to be analyzed is the effect of the processing time on RT packets delay, as shown from fig.3, the changes of the RT delay remain very small (almost zero) for values of Processing time below Arrival Time. And the values are relatively small (since all packets are served), a high transition (down to up) will occur, at the period where both processing time and arrival-time are equal, then the changes will be small until it reaches steady state maximum value, where all of the packets had been delayed the maximum delay.

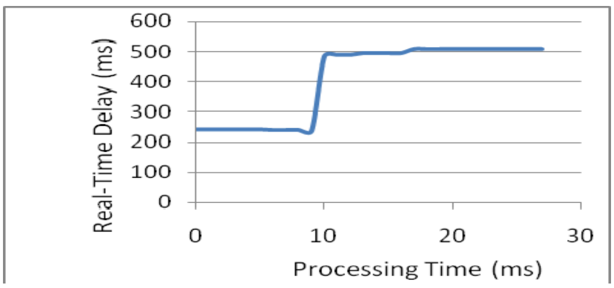

Fig.3. The effect of processing-time on RT delay

Another factor that affects the scheduling process is the arrival-time of the packets themselves, by fixing the RT deadline and the processing Time of the data packets, the system was executed for different values of the Time periods between packets, in this scenario we fixed the processing-time to be 10 ms, and the RT deadline to be 15 ms.

The first data to be analyzed is the effect of the Arrival-Time on Miss Ratio, as shown from Fig.4, the miss ratio will remain very high (almost 100%) for values of time-periods below the processing time, a high transition (up to down) will occur (near 0%) at the period where both processing time and time-period are equal, then the changes will be very small until it reaches 0% miss ratio.

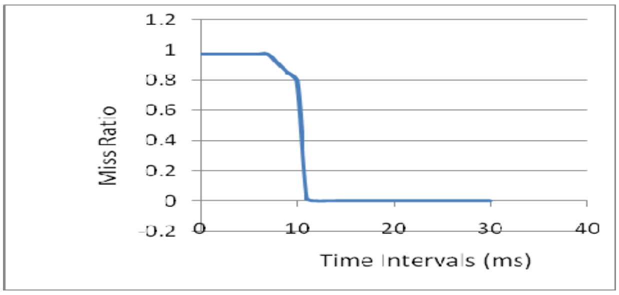

Fig.4. The effect of time-intervals on miss-ratio

The second data to be analyzed is the effect of the arrival time on total packets delays, as shown from Fig.5, an approximate linear relationship with a negative slope is appeared for values of time-periods below processing time, then the changes will be very small (almost zero) after that value of the processing-time (equal to time-interval).

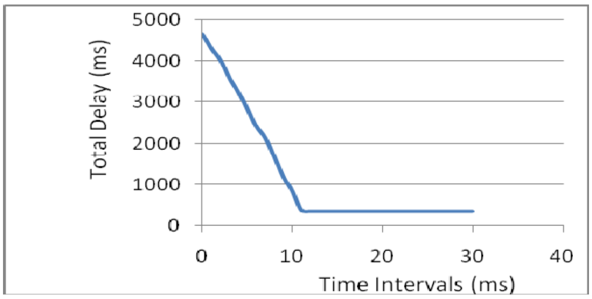

Fig.5. The effect of time-intervals on total-delays

The third data to be analyzed is the effect of the Arrival-Time on RT packets delays, as shown from Fig.6, the changes of the RT delay remain very high (maximum) for values of time-periods below Processing Time, and the values are relatively high (since most packets are dropped), a high transition (up to down) will occur, at the period where both processing time and time-periods are equal, then the changes will be small (almost zero).

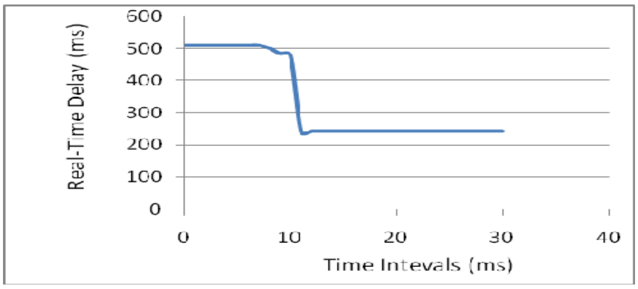

Fig.6. The effect of time-intervals on RT Delays

B. EDF scheduling algorithm

In the simulation process of this scheduler we put the EDF scheduler in the same conditions of the FCFS scheduler, so the first factor that affects the QoS requirements is the processing time of the scheduler, so by fixing the RT deadline and the time intervals between the data packets, the system was executed for different values of the processing time, in this scenario we have fixed the RT deadline to be15 ms, and the time periods to be 10 ms.

As shown from Fig.7, the miss ratio will remain very small (almost zero) for values of processing time below time-period, then an approximate linear relationship with a positive slope is appeared for values of processing-time above the time-period, then the changes will be very small until it reach 100% miss ratio.

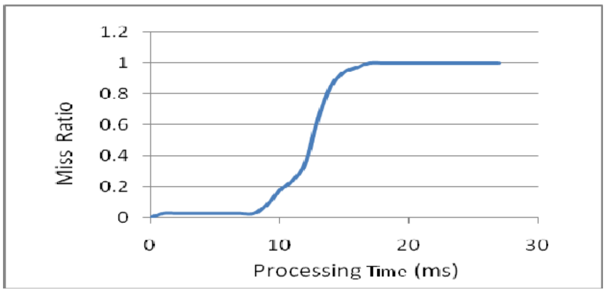

Fig.7. The effect of processing-time on miss ratio

The second data to be analyzed is the effect of the processing time on total packets delays, as shown from fig.8 the changes of the total delay remain relatively very small for values of processing time below arrival Time, an approximate linear relationship with a large positive slope is then appears after that value of the processing-time (equal to arrival-time).

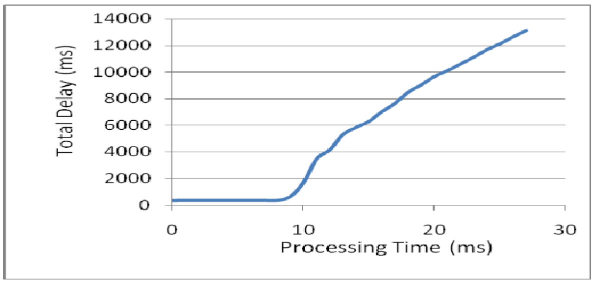

Fig.8. The effect of processing-time on total-delays

The third data to be analyzed is the effect of the processing time on RT packets delay, as shown from fig.9, the changes of the RT delay remain very small (almost zero) for values of Processing time below Arrival Time. And the values are relatively small (since all packets are served), an approximate linear relationship with a positive slope is then appears after that value of the processing-time, then the changes will be small until it reaches steady state maximum value, where all of the packets had been delayed the maximum delay.

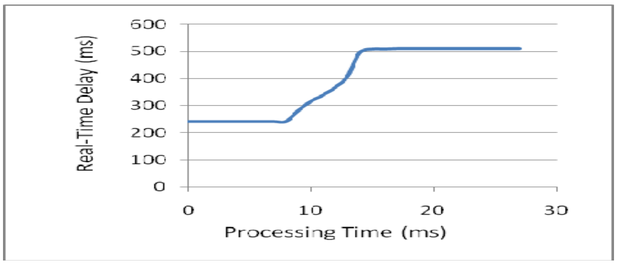

Fig.9. The effect of processing-time on RT delays

Another factor that affects the scheduling process is the arrival-time of the packets themselves, by fixing the RT deadline and the processing Time of the data packets, the system was executed for different values of the Time periods between packets, in this scenario we fixed the processing-time to be 10 ms, and the RT deadline to be 15 ms.

The first data to be analyzed is the effect of the Arrival-Time on Miss Ratio, as shown from Fig.10, the miss ratio will remain very high (almost 100%) for values of time-periods below the processing time, an approximate linear relationship with a negative slope is appeared while we still below it, then the changes will be very small until it reaches 0% miss ratio.

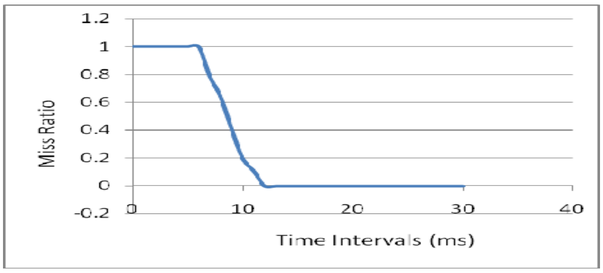

Fig.10. The effect of time-intervals on miss-ratio

The second data to be analyzed is the effect of the arrival time on total packets delays, as shown from Fig.11, an approximate linear relationship with a negative slope is appeared for values of time-periods below processing time, then the changes will be very small (almost zero) after that value of the processing-time (equal to time-interval).

The third data to be analyzed is the effect of the Arrival-Time on RT packets delays, as shown from Fig.12, the changes of the RT delay remain very high (maximum) for values of time-periods below Processing Time, and the values are relatively high (since most packets are dropped), an approximate linear relationship with a negative slope is appeared while we still below that point, then the changes will be small (almost zero).

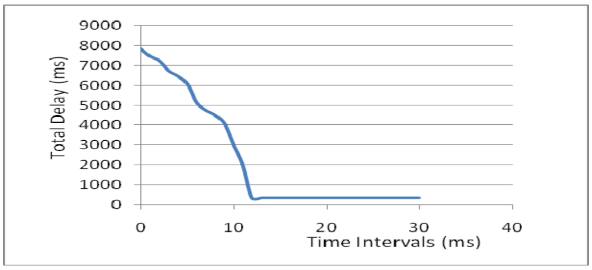

Fig.11. The effect of time-intervals on total delays

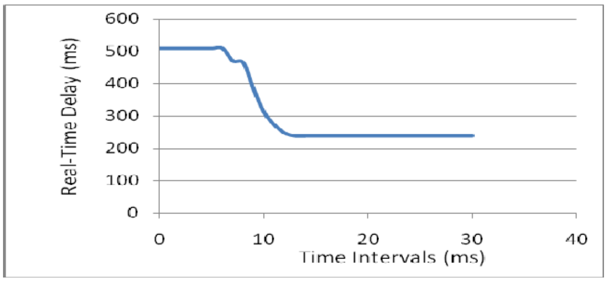

Fig.12. The effect of time-intervals on RT delay

C. *Comparison study*

By comparing the effect of the processing time on the miss ratio in both FCFS and EDF schedulers for the same conditions of time intervals and deadline, we can see from fig.1 the high transition (down to up) that occur (near 100%) at the period where both processing time and arrival-time are equal in the FCFS scheduler, but if we compare it to fig.7 we can see an approximate linear relationship with a positive slope is appeared for values of processing-time above the time-period in EDF scheduler. In this case if we take the critical point where both processing time and time intervals are equal to 10 ms, the FCFS scheduler will give us a miss ratio equal to 88% while EDF scheduler result is 17.6%, so the EDF performance will be:

$$\text{Performance} = 88\% / 17.6\% = 5$$

This means that the EDF is better than the FCFS by a factor of 5 (5 times better).

The second QoS requirement is the real-time delay, by comparing the effect of the processing time on the real-time delay in both FCFS and EDF schedulers for the same conditions of time intervals and deadline, we can see from fig.3 the high transition (down to up) that occur, at the period where both processing time and arrival-time are equal in the FCFS scheduler, but if we compare it to fig.9 we can see an approximate linear relationship with a positive slope is appeared for values of processing-time above the time-period in EDF scheduler. In this case if we take the critical point where both processing time and time intervals are equal to 10 ms, the FCFS scheduler will give us a real-time delay equal to 485 ms while EDF scheduler result is 315 ms, so the EDF performance will:

$$\text{Performance} = 485 / 315 = 1.54$$

This means that the EDF is better than FCFS by a factor of 1.54 (1.54 times better).

If we look again to the second factor that affects the QoS requirements of the system which is the time-periods, and if we compare the effect of the time intervals on the miss ratio in both FCFS and EDF schedulers for the same conditions of processing time and deadline, we can see from fig.4 that the miss ratio will remain very high (almost 100%) for values of time-periods below the processing time, and the high transition (up to down) will occur only (near 0%) at the period where both processing time and time-period are equal in FCFS scheduler, but if we compare it to fig.10 in EDF we will see that the miss ratio will remain very high (almost 100%) for values of time-periods below the processing time, and an approximate linear relationship with a negative slope is appeared while we still below it, then the changes will be very small until it reaches 0% miss ratio. In this case if we take the critical point where both processing time and time intervals are equal to 10 ms, the FCFS scheduler will give us a miss ratio equal to 79% while EDF scheduler result is 20%, so the EDF performance will be:

Performance = 79% / 20% = 3.95

This means that the EDF is better than the FCFS by a factor of 3.95 (3.95 times better).

The second QoS requirement is the real-time delay, by comparing the effect of the time-intervals on the real-time delay in both FCFS and EDF schedulers for the same conditions of processing time and deadline, we can see from fig.6 that the changes of the RT delay remain very high (maximum) for values of time-periods below Processing Time, and the values are relatively high (since most packets are dropped), and high transition (up to down) will occur, at the period where both processing time and time-periods are equal, then the changes will be small (almost zero) in FCFS, but if we compare that with fig.12 in EDF we can see that the RT delay remain very high (maximum) for values of time-periods below Processing Time, and the values are relatively high (since most packets are dropped), an approximate linear relationship with a negative slope is appeared while we still below that point, then the changes will be small (almost zero). In this case if we take the critical point where both processing time and time intervals are equal to 10 ms, the FCFS scheduler will give us a real-time delay equal to 479 ms while EDF scheduler result is 312 ms, so the EDF performance will:

Performance= 479 / 312 = 1.54

This means that the EDF is better than FCFS by a factor of 1.54 (1.54 times better).

One of the advantages of the FCFS scheduler over the EDF scheduler is that it's better to serve the best-effort traffic, as we can see from fig. 2 in the FCFS scheduler that the effect of the processing time on total packets delays is less than it in EDF scheduler where the effect is shown in fig.8, it's obvious that the positive slope of the linear approximation appears after the value where processing-time equal to arrival-time in FCFS scheduler is smaller than it in EDF scheduler. In this case if we take the critical point where both processing time and time intervals are equal to 10 ms, the FCFS scheduler will give us a total delay equal to 980 ms while EDF scheduler result is 1724 ms, so the FCFS performance will:

Performance= 1724 / 980 = 1.76

This means that the FCFS is better than EDF by a factor of 1.76 (1.76 times better). This result is expected since the total delay is a summation of both RT delay and NRT delay, and because the EDF scheduler will postpone the serving process of the NRT packets in the case of RT packet existence, then the NRT delay will be higher than it in FCFS, the small RT delay provided by EDF will not balance the problem since RT packets will not wait more than its deadline rather than the NRT packets which can wait for unlimited time.

REFERENCES

[1] Joel Goossens and Sanjoy Baruah and Sanjoy Baruah and Shelby Funk and Shelby Funk. Real-time Scheduling on Multiprocessors. in Proceedings of the 10th International Conference on Real-Time System, pages 189--204, 2002.

[2] Jayant R. Haritsa and Miron Livny and Michael J. Carey. Earliest Deadline Scheduling for Real-Time Database Systems. In Proceedings of the 12th Real-Time Systems Symposium, pages 232--242, 1991.

[3] Lorente, J.L. and Palencia, J.C. An EDF Hierarchical Scheduling Model for Bandwidth Servers. Embedded and Real-Time Computing Systems and Applications, 2006. Proceedings. 12th IEEE International Conference on, pages 261-266, 2006.

[4] J. Santoso and G.D. van Albada and B. A. A. Nazief and P.M.A. Sloot. Hierarchical Job Scheduling for Clusters of Workstations. 2000.

[5] XiPeng Xiao. The elusive QoS: what is missing?. Next Generation Internet Design and Engineering, 2006. NGI '06. 2006 2nd Conference on, pages 1 pp.-xv, 2006.

[6] Zhao, W. and Stankovic, J.A. Performance analysis of FCFS and improved FCFS scheduling algorithms for dynamic real-time computer systems. Real Time Systems Symposium, 1989., Proceedings., pages 156-165, 1989.

[7] Peifang Zhou and Yang, O.W.W. Scalability and QoS guarantee in IP networks. Computer Communications and Networks, 1999. Proceedings. Eight International Conference on, pages 427-433, 1999.

[8] Zhu, H.F. and Lehoczky, J.P. and Hansen, J.P. and Ragunathan Rajkumar. Diff-EDF: a simple mechanism for differentiated EDF service. Real Time and Embedded Technology and Applications Symposium, 2005. RTAS 2005. 11th IEEE, pages 268-277, 2005.