# Quantized EDF Scheduling in a Stochastic Environment *

Jeffery P. Hansen
Institute for Complex Engineered Systems
Carnegie Mellon University
Pittsburgh, PA 15213
Email: *hansen@cmu.edu*

Haifeng Zhu
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
Email: *haifengz@ece.cmu.edu*

John Lehoczky
Dept. of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213
Email: *jpl@stat.cmu.edu*

Ragunathan Rajkumar
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
Email: *raj@ece.cmu.edu*

## Abstract

*In this paper, the effects of quantization on EDF (Earliest Deadline first) queues with a stochastic traffic model are presented. In a quantized EDF queue, jobs have limited fidelity in representing their relative deadlines. It is shown that most of the benefits of full EDF can be achieved with relative deadlines expressed in as little as three bits. The implications of this are that QoS networking technologies such as DiffServ can be used to implement near EDF performance while still using no more than the standard set of bits in DS/TOS field.*

## 1 Introduction

Application capabilities and user demands are rapidly increasing the resources required in distributed computing systems. Even when large amounts of resources are available, inefficient use of those resources can lead to unacceptable application performance. One way to ensure application demands are met is through the use of resource management [1][5][2][3][8][7].

One important dimension of application quality is timeliness. EDF (Earliest Deadline First) scheduling focuses on providing resources to real-time tasks. Each task has a deadline by which the task must be completed, and tasks are serviced in order of increasing deadlines. EDF is useful in critical real-time systems, high-speed networks and other applications. A pure EDF scheduling algorithm assumes an infinite number of priority levels. In practice, this is not always feasible, especially in small real-time devices or in networking.

One way to make EDF more practical for use in networking is to use a fixed set of possible deadlines where tasks request the next smaller deadline from the set than that of their actual deadline. Each deadline can then be assigned a code, and this code used to indicate the deadline of a task, or in a packet to indicate delivery requirements. We call this approach quantized EDF scheduling. In this paper we show that in the context of a firm deadline system (i.e., a small fraction of missed deadlines are allowed), that near EDF performance can be achieved with only a 3-bit code to represent the deadline.

## 2 Real-Time Queueing Theory

Hard real-time system scheduling theory usually provides hard guarantees in which all task deadlines are met with certainty. This requirement is forced to be achieved even under the worst cases, resulting in substantial a waste of system resources. The advantage of these mechanisms is a 100% guarantee of performance, while its disadvantage is low utilization. In practice, average case performance is usually much better than worst case performance. This is especially true when guarantees must be made for variable bit-rate flows.

*This work was supported by the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant N00014-01-1-0576

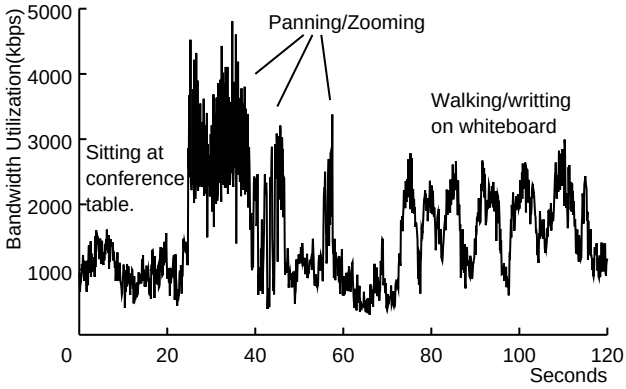


**Figure 1. Bandwidth consumption of a video-conference flow**

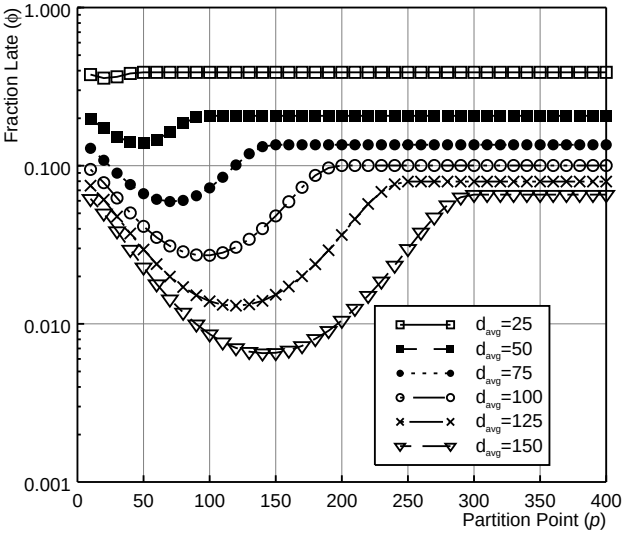


**Figure 2. Effect of Mean Deadline on Optimal Partition for Uniform Deadlines ($\rho = 0.96$, $d_{\min} = 5$)**

For example, Figure 1 shows the bandwidth consumed by a video-conference at 30 frames per second video-conference. The data was generated by recording an actual video-conference on video tape and then playing that tape back through a video capture card on a PC running vic[9] under the H.261 format. Due to the compression algorithms used and variations of the amount of motion in the video, the bandwidth required can range from 0.5Mbps to 5Mbps. In order to provide hard deadline guarantees on such a flow, it would be necessary to design for the worst case. A better approach is specify a fraction of deadlines $\phi_{\max}$, which we can tolerate. By allowing a small fraction of deadlines to be missed (e.g., 1%) system utilization can be increased dramatically.

Real-time queueing theory has been proposed as a technique to analyze a queueing system in a stochastic environment[6]. While traditional queueing analysis usually provides average performance estimation, it does not focus on timing analysis. Real-time queueing theory extends heavy traffic queueing theory utilizing the fact that the number of tasks in the queue behaves like reflected Brownian motion under heavy traffic. It can be used to estimate the lead-time profile of tasks in a queue from the arrival distribution, the service distribution, the deadline distribution and the queueing policy. The lead-time of a task is simply the time remaining until the deadline expires. A negative lead-time indicates a task deadline has already expired, while a positive lead-time indicates the deadline has not yet expired. The lead-time profile is the CDF (Cumulative Distribution Function) of lead-time for the tasks in the queue. The lead-time profile for the tasks in a queue depends on the number of tasks in the queue. Exit lead-time distribution and fraction of missed deadlines can be computed from the queue-length dependent lead-time profiles and the queue-length distribution obtained through standard queueing theory.

While real-time queueing theory generally assumes a stochastic traffic model, real traffic may not always be easily modeled as a simple stochastic process as was shown in Figure 1. In work on probabilistic bandwidth allocation[4], network traffic generated by a video stream was modeled as a Markov modulated Poisson process with states representing activity levels such as quiet, medium motion and high motion. A similar model could be applied to real-time queueing theory analysis and to the Q-EDF analysis presented in this paper with task/packet interarrivals being treated as "locally exponential" within a mode.

## 3 Quantized EDF Scheduling

In pure EDF scheduling, a task arriving at time $t_c$ with a relative deadline of $d$ has an absolute deadline of $t_c + d$. In our model of quantized EDF (Q-EDF) scheduling, the relative deadline is modified with a quantization function $Q(d)$ which returns the largest value from a set $D = \{d_1, d_2, ..., d_n\}$ which is less than or equal to $d$. In this paper we will call each of these possible relative deadline values a bin. Note that these bins are merely a convenience for analysis purposes and an actual implementation of a Q-EDF queue need not treat them as separate buffers. This model reflects the fact that in a network, while there may be a limited number of bits available to encode a deadline (e.g.,

such as in the DS/TOS field of an IP packet), relatively higher time resolution may be available on the router. We call the value $t_c + Q(d)$ the effective deadline of the task. Insertion into the Q-EDF queue are similar to a standard EDF queue except the effective absolute deadline is used rather than the actual absolute deadline. Furthermore, tasks having the same effective deadline will be inserted in FIFO order.

Since all tasks have an effective deadline smaller than their actual deadlines, if a task meets its effective deadline, it will also meet its actual deadline. However, quantization may cause lower priority tasks to be serviced before higher priority tasks leading to priority inversions resulting in a higher fraction of missed deadlines under Q-EDF as compared to pure EDF. In the extreme case of a single bin, the result is a plain FIFO queue, and the other extreme of an infinite number of bins the result is a pure EDF queue.

Simulation is used to compare quantized EDF queues with FIFO and pure EDF queues. The simulator was written in C and uses event-based simulation with each simulation run for $10^7$ dequeue events. Different seed values resulted in little variation between runs demonstrating that the simulation length was adequate to achieve convergence. In all of our experiments, the arrival and service times are considered to be exponentially distributed with $\lambda$ fixed at 1. The traffic intensity $\rho$ is varied by changing the average task service time $1/\mu$. The primary metric used to evaluate an operating point is the fraction of tasks $\phi$ which are processed late. In most cases, we would choose to have no more than 1% missed deadlines.

Both uniformly and exponentially distributed deadlines are considered. Deadlines are characterized by a minimum deadline $d_{\min}$ and average deadline $d_{\text{avg}}$. If $\mathbf{u}$ and $\mathbf{e}$ are respectively unit uniform and unit exponential random numbers, then uniformly distributed deadlines are generated as $d_{\min} + (2d_{\text{avg}} - d_{\min})\mathbf{u}$ while exponentially deadlines are generated as $d_{\min} + (d_{\text{avg}} - d_{\min})\mathbf{e}$. Note that in a real network, the relative deadlines for the packets in a given flow are typically identical. The deadline distribution modeled here, represents the deadlines in an aggragate of a large number of flows.

In most of the experiments, we concentrate on the heavy traffic case (i.e., $\rho \geq 0.95$) on the assumption that if we can handle the heavy traffic case, the missed deadline rate $\phi$ for lower traffic intensities will necessarily be lower.

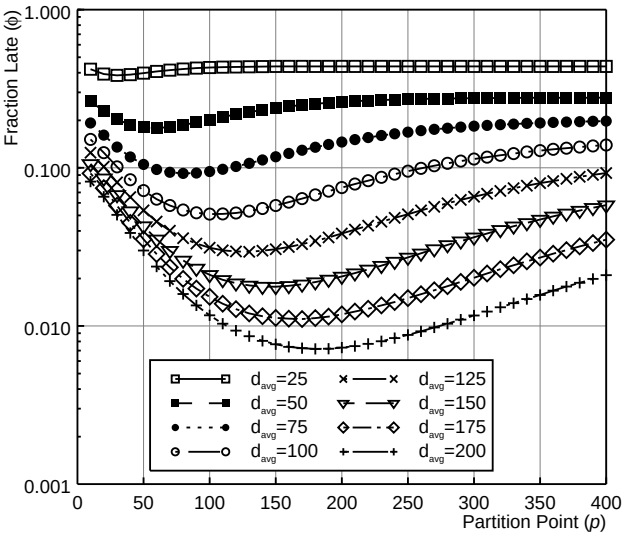


**Figure 3. Effect of Mean Deadline on Optimal Partition for Exponential Deadlines** ($\rho = 0.96$, $d_{\min} = 5$)

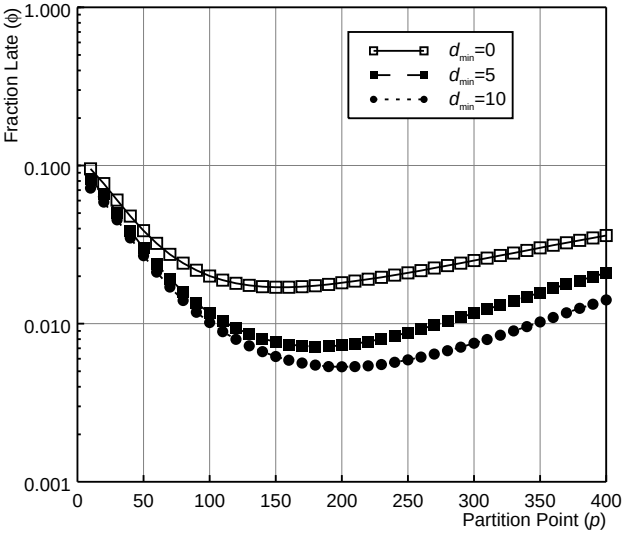


**Figure 4. Effect of Minimum Deadline on Optimal Partition for Exponential Deadlines** ($\rho = 0.96$, $d_{\mathbf{avg}} = 200$)

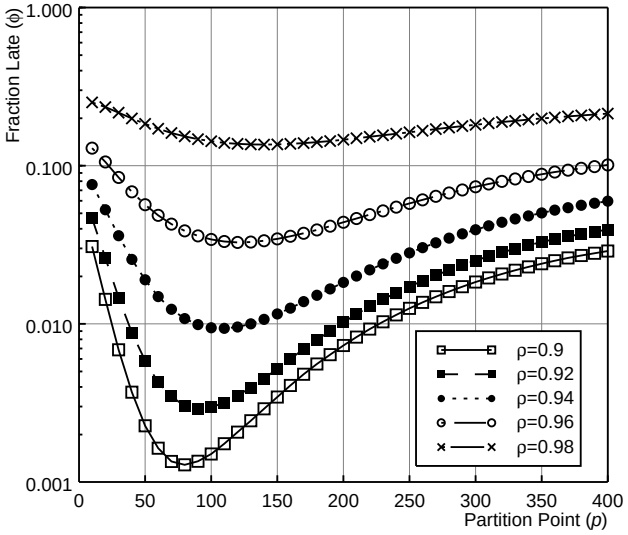


**Figure 5. Effect of Traffic Intensity on Optimal Partition for Exponential Deadlines ($\rho = 0.96$, $d_{\mathbf{avg}} = 120$)**

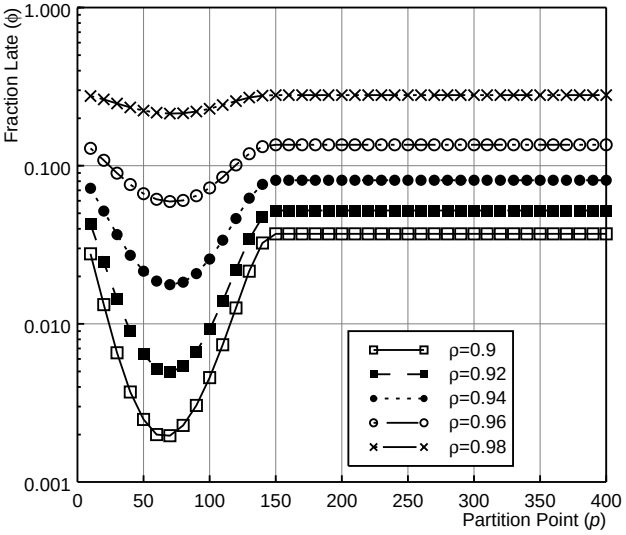


**Figure 6. Effect of Traffic Intensity on Optimal Partition for Uniform Deadlines ($\rho = 0.96$, $d_{\mathbf{avg}} = 75$)**

## 4 2-Bin Quantization

The simplest form for a Q-EDF queue is a two-bin queue characterized by a partition point $p$. The effective relative deadline is either 0 if the actual relative deadline is less than $p$, and $p$ if the actual relative deadline is greater than $p$.

A minimum deadline is necessary to prevent tasks from being late as soon as they are inserted into the queue. For example, if the average service time of a queue was $32\mu s$, but a task with a deadline of $1\mu s$ was inserted into the queue, the deadline would nearly always be missed no matter what queueing discipline we used. This effect is shown in Figure 4. This figure shows the fraction of actual deadlines missed as the partition point $p$ is varied from 10 to 400 under heavy traffic ($\rho = 0.96$). The deadlines are exponentially distributed with and average deadline of $d_{\text{avg}} = 200$ and minimum deadlines ranging from 0 to 10. As can be seen in the figure, there is a large decrease in missed deadlines as the minimum deadline is increased from 0 to 5, but a much smaller effect going from 5 to 10. For this reason, we choose a minimum deadline of 5 in the remainder of the experiments in this paper. The practical meaning of this is that the minimum deadline should be about five times the average service time. For example, assume an average packet size of 400 bytes on a 100Mbps link. This results in an average packet transmission time of about 32 $\mu s$ and thus a minimum feasible deadline of about 160 $\mu s$.

The best partition point $p$ for a particular deadline distribution depends on the mean of the distribution. Figure 2 shows the heavy traffic fraction of actual deadlines missed as a function of the partition point for uniformly distributed deadlines with means ranging between 25 and 150. As can be seen in the figure, the optimal partition point is very near the mean, while as expected the fraction missed increases as the mean deadline decreases. A similar result is obtained for exponential deadlines (Figure 3), although with an elongated tail. Note also that the fraction of deadlines missed for exponential deadlines is higher than that for uniform deadlines with the same mean due to the increased number of short deadlines in the exponential case.

As the traffic intensity decreases from the heavy traffic case, the optimal partition point for exponentially distributed deadlines drops below the mean (Figure 5). This effect is much less pronounced for uniform deadlines (Figure 6). In both cases, while choosing a partition point equal to the mean may not be optimal, for traffic intensities down to at least 0.9, the deadline mean makes an acceptable choice.

The results of the previous experiments suggest a methodology for choosing the partition point based on system requirements. Given any two of the mean deadline $d_{\text{avg}}$, the maximum allowed fraction of late customers $\phi_{\max}$ and the traffic intensity $\rho$, the other parameters as well as the optimal partition point $p$ can be found. We first assume that $d_{\text{avg}} = p$ as suggested in

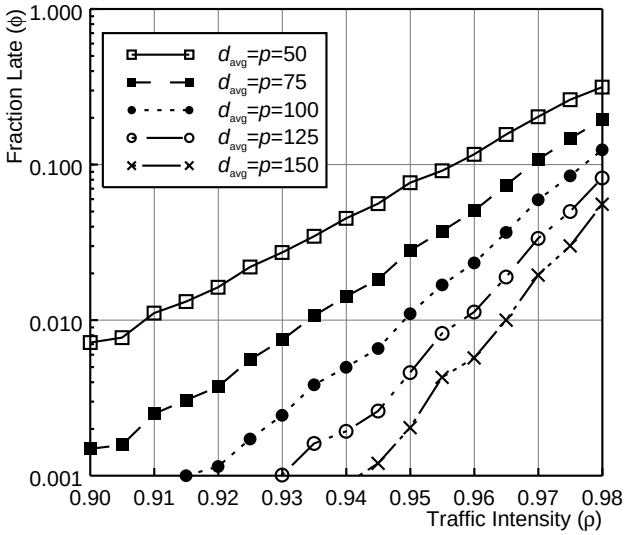


**Figure 7. Fraction Late for Uniform Deadlines ($d_{\min} = 5$)**

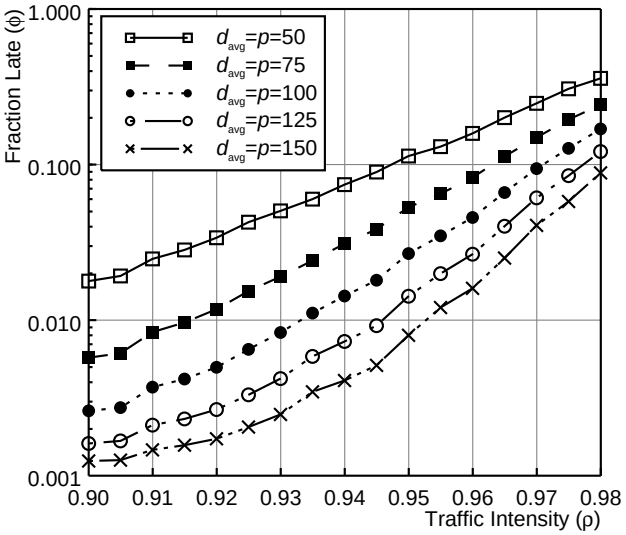


**Figure 8. Fraction Late for Exponential Deadlines ($d_{\min} = 5$)**

the previous experiments. We then ran a series of simulations for various values of $p$ (with $d_{\text{avg}} = p$) with the traffic intensity plotted against the fraction late. For example, referring to Figure 7 for uniform deadlines, if we wish to have no more than 1% of deadlines missed while operating at a traffic intensity of 0.94, we find the corresponding point on the graph and identify the curves closest to it. In this case, the point lies between the plotted $d_{\text{avg}} = p = 75$ and $d_{\text{avg}} = p = 100$ curves. By interpolation we can determine that the minimum mean deadline $d_{\text{avg}}$ that can be tolerated under these conditions is about 85 and thus the partition point $p$ should also be about 85. A similar set of curves can be generated for exponential deadlines (Figure 8).

## 5 Multi-Bin Quantization

In general, a multi-bin quantized EDF queue is characterized by its set of quantization points $d_i$ taken from a set $D$. In this paper, we will restrict our study to "uniform" bin quantization and "log" bin quantization. Both uniform and log bin quantization schemes are characterized by a minimum deadline $d_{\min}$, a maximum deadline $d_{\max}$, and the number of bins $n$. In the uniform case, the bin partition points $d_1, d_2, ..., d_n$ are at $d_i = d_{\min} + (i-1)\frac{d_{\max} - d_{\min}}{n}$. In the exponential case the bin partition points are at $d_i = d_{\min} b^{i-1}$ where $b = \sqrt[n]{\frac{d_{\max}}{d_{\min}}}$.

The selection of $d_{\min}$ was addressed in the previous section and relates to the smallest reasonable deadline that can be requested of the system. In the experiments reported here, we choose $d_{\min} = 5$. The selection of $d_{\max}$ determines the largest deadline that can be distinguished. While choosing a very large value increases the range of deadlines which can be expressed, it also makes all of the bins proportionally larger. In this paper we choose $d_{\max} = 1000$ to maintain reasonable bin sizes while allowing a reasonable range of deadline values. $d_{\max}$ should be chosen so that there are few or no tasks with initial deadlines greater than $d_{\max}$.

The effect of increasing the number of uniformly spaced bins $n$ on the fraction of late packets $\phi$ for uniformly distributed deadlines is shown in Figure 9. Note that the $n = 1$ case corresponds to a FIFO queue. The curve for pure EDF is also shown indicating the upper bound for performance by the quantized EDF queue. As is evident in the figure, there is a large increase in performance (in terms of the smallest average deadline that can be supported for a specific value of $\phi$) going from $n = 1$ to $n = 2$, but after $n = 16$ there is relatively little benefit of increasing the number of bins. One aspect that is notable in the figure is that the $n = 1$ and $n = 2$ curves overlap up to an mean deadline value of about 250. Since the bins are uniformly spaced and $d_{\max} = 1000$, the first bin covers deadlines between 5 and 502. This means that with uniformly distributed deadlines for $d_{\text{avg}} \leq 251$, all of the tasks are placed in the first bin making it equivalent to FIFO. A similar effect is noticeable for larger values of $n$ although at smaller $d_{\text{avg}}$ values.

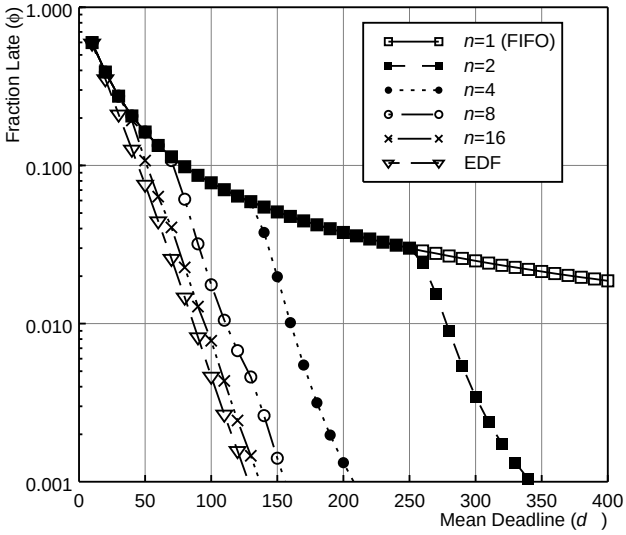


**Figure 9. Effect of Mean Deadline on Quantization for Uniform Bins and Uniform Deadlines** ($\rho = 0.95$, $d_{\min} = 5$, $d_{\max} = 1000$)

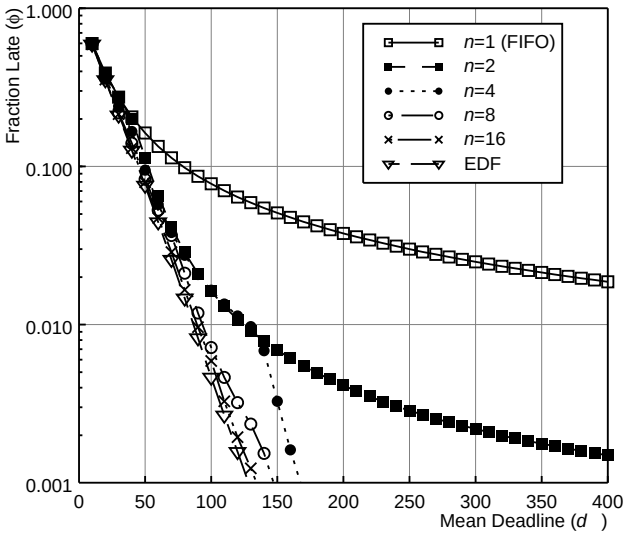


**Figure 10. Effect of Mean Deadline on Quantization for Log Bins and Uniform Deadlines** ($\rho = 0.95$, $d_{\min} = 5$, $d_{\max} = 1000$)

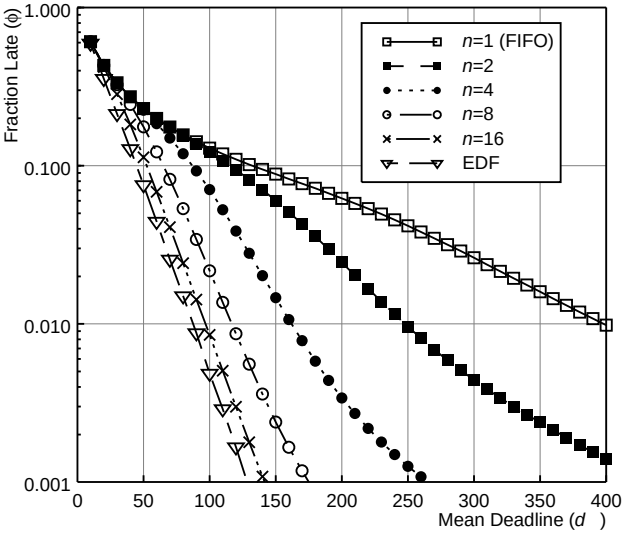


**Figure 11. Effect of Mean Deadline on Quantization for Uniform Bins and Exponential Deadlines** ($\rho = 0.95$, $d_{\min} = 5$, $d_{\max} = 1000$)

With log spaced bins, adding even a single bin (i.e., $n = 2$) results in performance comparable to $n = 8$ for the uniform bin case (Figure 10). As the number of bins is increased to $n = 8$, there is virtually no difference in performance between pure EDF and quantized EDF scheduling.

Performance curves for exponentially distributed deadlines with uniform bins and log bins are shown respectively in Figures 11 and 12. While there is some decrease in performance compared to the uniform deadline case due to the larger number of short deadlines, once the number of bins goes beyond $n = 16$ for the uniform bin case, or $n = 8$ for the log bin case performance is still comparable to pure EDF scheduling. We also see that just as with uniform deadlines, the log deadlines improve performance for exponentially distributed deadlines when compared to uniform bins.

We also investigated the effects of traffic intensity on the number of bins required as shown in Figure 13. In this experiment, the average deadline was held constant at $d_{\text{avg}} = 100$ (with exponential deadlines) and the traffic intensity was varied for different bin counts. From this experiment we see that even as traffic intensity varies (within the heavy traffic range) eight bins is sufficient to obtain performance comparable to pure EDF scheduling.

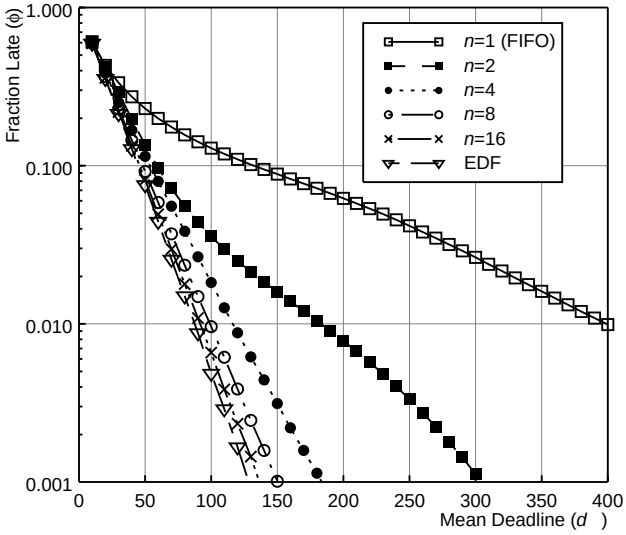


**Figure 12. Effect of Mean Deadline on Quantization for Log Bins and Exponential Deadlines ($\rho = 0.95$, $d_{\min} = 5$, $d_{\max} = 1000$)**

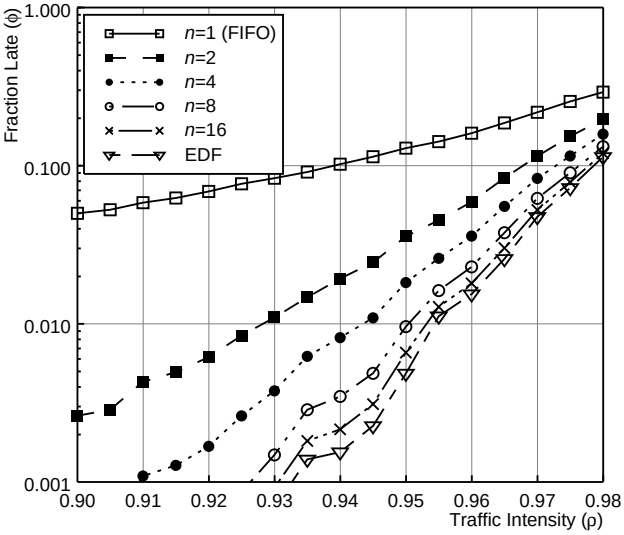


**Figure 13. Effect of Traffic Intensity on Quantization for Log Bins and Exponential Deadlines ($\rho = 0.95$, $d_{\min} = 5$, $d_{\max} = 1000$, $d_{\mathbf{avg}} = 100$)**

## 6 Conclusion

In this paper we have presented experimental results on the delay characteristics of quantized EDF queues. We have shown that in most cases 8 bins is sufficient to represent the deadline and obtain performance comparable to that of a pure EDF queue. This means that tasks can encode their deadline in only three bits. We have also shown that performance for log spaced bins is better than for uniformly spaced bins due to the fact that they provide more granularity for short deadlines.

## References


[1] P. Chandra, A. Fisher, C. Kosak, T. S. Eugene Ng, P. Steenkiste, E. Takahashi, H. Zhang, "Darwin: Resource Management for Value-Added Customizable Network Service", From *6th IEEE International Conference on Network Protocols (ICNP'98)*, October 1998

[2] C.L. Hoover, J.P. Hansen, P. Koopman, S. Tamboli, "The Amaranth Framework: Policy-Based Quality of Service Management for High-Assurance Computing", Accepted for publication in *International Journal of Reliability, Quality, and Safety Engineering*

[3] C.L. Hoover, J.P. Hansen, P. Koopman, S. Tamboli, "The Amaranth Framework: Probabilistic, Utility-Based Quality of Service Management for High Assurance Computing", From *Proceedings of the 4th IEEE International High-Assurance Systems Engineering Symposium (HASE '99)*, November 1999

[4] J.P. Hansen, H. Zhu and R. Rajkumar, "Probabilistic Bandwidth Reservation by Resource Priority Multiplexing", From *Proceedings of the 2001 Real-Time Applications Symposium*, July 2001

[5] G. Nutt, S. Brandt, A. Griff, S. Siewert, T. Berk, and M. Humphrey, "Dynamically Negotiated Resource Management for Data Intensive Application Suites", IEEE Transactions on Knowledge and Data Engineering, 12(1):78-95, January/February 2000.

[6] J.P. Lehoczky, "Real-Time Queueing Theory", In *Proceedings Real-Time Systems Symposium*, pg. 186-195, December 1996

[7] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, "RSVP: A New Resource ReSerVation Protocol", From *IEEE Network*, vol. 7, no. 5, p. 8-18, September 1993

[8] L.R. Welch, B. Shirazi, B. Ravindran, C. Bruggeman, "DeSiDeRaTa: QoS Management Technology for Dynamic, Scalable, Dependable, Real-time Systems", 15th IFAC Workshop - Distributed Computer Control Systems (DCCS '98), Sept. 1998, pp. 7-12.

[9] UCB/LBNL Video Conferencing Tool, http://www-nrg.ee.lbl.gov/vic