A Novel Approach for Real Time Flows Scheduling

S. Ould Cheikh El Mehdi[1], W. Fawaz[1,2], K. Chen[1]
(1) L2TI Lab., Institut Galilée, University Paris 13
99, Av. J.B. Clement, 93 430 Villetaneuse, France
saadbouh@l2ti.univ-paris13.fr, {fawaz, kchen}@galilee.univ-paris13.fr
(2) Dept. of Electrical and Computer Engineering
Lebanese American University, P.O. Box 36, Byblos, Lebanon

***Abstract*— A plethora of packet-scheduling algorithms have been proposed in the literature in order to meet the stringent time constraints of real time flows at an IP router level. In this regard, the so-called EDF algorithm attracted special attention since it is prominent for optimally managing flows with strict time constraints. However, EDF is complex and expensive as far as implementation is concerned, especially when compared with the standard FIFO algorithm. As a main contribution in this paper, we therefore propose a novel hybrid scheduling approach, which combines the optimality of EDF and the simplicity of FIFO. This approach allows reducing EDF's implementation complexity while making efficient use of its optimal flow management. Our simulation results underline the benefits behind such a proposal.**

***Index Terms*: - Real-time Scheduling, Quality of Service (QoS), Earliest Deadline First (EDF), Implementation optimization.**

I. Introduction

The Internet technology is becoming a *de facto* standard for almost all kind of communications, not only in wide area networks, but also in more restrictive areas. Actually, more and more standard equipments and applications, as well as development manpower, are becoming IP-oriented.

Internet has long been limited by the type of service provided to end users, and which relies on the best effort concept basis: the network holds the sole promise of doing its best regarding packets' delivery to their destination, and no more guarantees are offered to the end users' traffic. This best effort type of service is quite suitable for the so-called *elastic applications* (basically TCP traffics) which may tolerate delay variations while compensating for eventual packet losses through retransmissions. In this regard, since the network offers a minimal service, ensuring better end to end service conditions is dealt with at the transport layer of end users' applications. These applications were thus content with the minimal service context of the Internet.

However, the perpetual development of the Internet is boosting the creation of new types of applications, such as multimedia applications (voice over IP (VoIP), video conferences,...), and whose requirements can not be satisfied through the best effort service context. Indeed, these emerging applications are presenting stringent real time constraints through the strict delay and/or bandwidth requirements, which are needed by their generated traffic flows. The strictness degree of such requirements varies from one type of application to another. Meeting these various constraints implies the need to provide the different applications with differentiated Quality of Service (QoS) levels, which are adapted to their needs.

In retrospect, several pioneer works have been conducted in the literature to deal with the QoS issues in packet routed multimedia networks. These activities led particularly to the proposition of different packet scheduling techniques, among which one can cite Weighted Fair Queuing (WFQ) [1][2] also known as Packet-by-Packet Generalized Processor-Sharing (PGPS), and its variants [3]. Such scheduling techniques are generally deployed within a multimedia network enforcing the IntServ or the DiffServ framework. The rationale behind these scheduling techniques lies in guaranteeing the bandwidth required by each data flow while bounding the response time of the data flow. Nonetheless, the response time resulting from the deployment of such scheduling techniques, referred to as *share-driven scheduling algorithms*, is function of both traffic burst size and the reserved bandwidth. This is especially true, since in the case of a bursty traffic, the end to end delay increases linearly with the increase of the maximal burst size. As a result, this may lead to the non-respect of real time packets' deadline. To cope with such a limitation, other scheduling techniques referred to as *deadline driven scheduling algorithms* are proposed for deployement in the network. In this article, our attention will be focused on one of such scheduling techniques, the so-called EDF (Earliest Deadline First) algorithm [4][5][6][7], which is widely known in the context of real time traffic scheduling. The main idea behind EDF is the following: to each task is associated a deadline, which indicates a kind of *timely varying priority*, or the maximum allowed waiting time. Tasks are served according to an increasing order of their associated deadline values.

EDF has been proven to be an optimal scheduling discipline in the sense that if a set of tasks is *schedulable* under any scheduling discipline (In other words, if the packets can be scheduled in such a way that all of their deadlines are met), then the set is also *schedulable* under EDF [8]. When applied to a networking case ([9],[10]): people propose to associate to each flow i, and at the level of a given router m along i's path, a local deadline value d_i^m. A certain number of theoretical studies ([11][12] and [8])proved that EDF is optimal with regard to several criteria (For instance, the percentage of unsatisfied deadlines, etc.). Furthermore, the implementation of EDF in a real network was the subject of numerous studies (cf [13][14][15][16] for some of the first proposals, and [17][18][19][20] for more recent works). However, this algo-

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

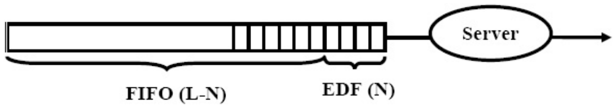

Fig. 1. The scheme of our hybrid EDF/FIFO Queue.

rithm suffers from a major drawback related to the additional cost resulting from packet classification overhead, and which becomes more pronounced when the queue length increases drastically. This is augmented with an efficiency concern in the case of heavy load.

In this article, attempting to reduce the cost induced by EDF while taking advantage of its optimal aspect, we propose a novel hybrid queue management approach, which combines both the EDF and the FIFO scheduling algorithms. Instead of using EDF for the scheduling of the whole queue content, our idea consists in limiting the usage of EDF to the scheduling of the first N packets; and the remaining packets are scheduled via the simple and fast FIFO algorithm. In the article [21], a similar algorithm that combines FIFO and Minimum Laxity (considered as an EDF variant) was proposed with a small N value. Our work may be viewed as a complementary effort regarding this idea of hybrid management FIFO/EDF, analyzing its performance through simulations and with a more realistic N value. We performed such simulations using the Network Simulator 2 (NS-2) [22], to which we added new extension modules related to our study.

The paper is structured as follows: in section 2, we propose and describe the hybrid queue management mechanism. In section 3, a simulation study is developed with the corresponding results to gauge the benefits of our proposal. Finally, Section 4 concludes this paper and proposes future issues.

II. A Novel Queue Management Mechanism

A. Rationale

This section introduces the proposed hybrid queue management approach that combines the EDF and FIFO algorithms. Recall first that the main objective behind such a combination is to reduce the overall cost as well as the complexity of the EDF algorithm. To achieve this objective, the application of EDF algorithm will be limited to a scope of up to N packets of the queue. As for the remaining packets, the FIFO algorithm is applied. There are *a priori* several possible ways (schemes) to combine the EDF and the FIFO queues. Two main configurations are respectively a) EDF at the head of the queue, and b) EDF at the tail of the queue. In [21], it has been proven that they are statistically equivalent. In this paper, we take the EDF at head variant. Figure 1 depicts such a hybrid queue of length L, which comprises two serial queues (EDF and FIFO). EDF, as stated previously, handles the first N consecutive packets (if any), while FIFO is applied to the $L - N$ remaining packets. This is the working model of this paper.

It is clear that this hybrid queue aims to achieve a complexity reduction without noticeable loss of performance, when compared to a pure EDF queue. Thus, N is a key parameter. The greater the value of N will be, the more we will be approaching the performance of a pure EDF queue. In fact, for $N = L$, the behavior (and so performance) of the hybrid queue will be identical to that of an EDF queue. On the other hand, the lower the value of N will be, the lesser the complexity induced by EDF classification will be. In Section III, quatitative studies obtained by simulation will be presented.

B. Functional description

1) Two operational modes: The hybrid queue may treat arriving packets according to different operational modes. We have chosen to study two operational modes, they are referred to as respectively the *normal* mode, and the *enhanced* mode:

- Under the *normal mode*, upon arrival of a new packet, if the EDF queue is not full, the packet is inserted into the EDF queue; otherwise (the EDF queue is full), it is put directly to the tail of the FIFO queue. The insertion cost, when the EDF queue is full (the total number of customer is greater than N), is thus $O(1)$ (the FIFO insertion operation).
- Under the *enhanced mode*, upon arrival of a new packet, its insertion into EDF queue is systematically tried. If the EDF queue is already full and the new one is to be inserted, the former last one of the EDF queue will be pushed into the head of the FIFO queue. If the new one is not to be inserted into EDF queue, then it is put to the tail of FIFO queue. Thus, insertion cost when the EDF queue is full is now $O(\log_2(N))$, plus the FIFO insertion (either at head or tail) operation (cost: $O(1)$).

The rationale of the *enhanced mode* is to get a behavior closer to the one obtained by a pure EDF, at the expense of a systematic insertion effort. Indeed, the additional cost of *enhanced mode* is $O(\log_2(N))$ but its behavior is clearly closer to the one of pure EDF than the *normal mode* . The two variants conserve nevertheless the same basic characteristic of the hybrid queue, which is the existence of a deterministic upper bound on insertion cost.

2) Algorithmic description: Hereafter we give a pseudo-algorithmic description of the hybrid queue with its two variants (*normal* and *enhanced* modes).

a) Queue definition and variables: We consider a serial FIFO/EDF hybrid queue as illustrated in Figure 1

- N is the length of the EDF queue, L the total length of this hybrid queue;
- L_1 (rep. L_2) is the number of packets in the EDF (resp. FIFO queues), initially $L_1 = L_2 = 0$.

There are two concurrent operations: new packet insertion and packet service. For the sake of simplicity, the newly arrived packet is denoted P_a, and its deadline by D_a. In addition, the last packet of the EDF queue is denoted P_l and its deadline by D_l. The operation *insertion into EDF (resp. FIFO)* means implicitly insertion according to the EDF (resp. FIFO) scheduling discipline.

b) New packet insertion: Upon arrival of a new packet (P_a), the insertion is performed either in `Normal` or `Enhanced` mode.

- IF (`mode=Normal`) THEN
  - IF $L_1 < N$ THEN insert P_a into EDF, $L_1 := L_1 + 1$

- ELSE insert P_a into FIFO, $L_2 := L_2 + 1$

- ELSE (in this case `mode=Enhanced`)
  - IF $L_1 < N$ THEN insert P_a into EDF, $L_1 := L_1 + 1$
  - ELSE IF ($D_a < D_l$) THEN move P_l (the last packet of the EDF queue) to the head of FIFO, insert P_a into EDF, $L_2 := L_2 + 1$
    * ELSE insert P_a into FIFO, $L_2 := L_2 + 1$
- Check for rejection (of the last one of FIFO), if any :
  - IF $L_1 + L_2 > L$ THEN Reject the last one of FIFO, $L_2 := L_2 - 1$

c) Packet service: Upon departure (service complexion) of a packet,

- the new first packet (if any) of EDF queue will be served without delay
- IF FIFO is not empty, THEN the first one of FIFO is fetched to be inserted into EDF queue (according to its deadline) and $L_2 := L_2 - 1$,
- ELSE $L_1 := L_1 - 1$

III. Hybrid Queue Performance Evaluation: Simulation Study

The analytical modeling of this hybrid queue is rather difficult. As a first attempt, we carried out simulation studies using the *network simulator 2* package (*NS-2*) [22], to which we added specific modules implementing the two variants of our hybrid queue.

In this section, we present our investigation, through simulations, on the performances of the two hybrid queue variants proposed in the previous section, in terms of deadline respect percentage. The main objective is to evaluate the benefits of the hybrid queue in comparison with pure EDF, and pure FIFO queues.

Recall that within the hybrid FIFO/EDF queue of length L, the EDF algorithm is applied to the first N packets while FIFO is enforced for the remaining packets ($L - N$ consecutive packets). Doing so, the complexity related to the EDF algorithm classification is reduced while maintaining to some extent the advantages of the EDF algorithm (i.e., optimal packets' deadline respect). However, to ensure a reasonable trade-off between performance improvement and complexity reduction, an appropriate value needs to be assigned for the N parameter. Achieving better performances in terms of deadline respect requires increased values of N, in which case we approach the performances of an EDF queue. Conversely, reducing the hybrid queue's complexity necessitates a reduced value of the N parameter. Building on this analysis, it is clear that N is a key parameter in our simulations. Therefore, simulation results for different values of N will be presented later on. But still, we focus on the lowest value of N realizing near pure EDF performances.

A. Simulation Scenario

We consider a network topology comprising 3 traffic sources, sharing equally a 2 Mb/s link. Packets have fixed size of 150 bytes. The link is managed as respectively a pure EDF, a pure FIFO, or one of the proposed hybrid queue variants. The buffer size is set to 80 packets ($L = 80$).

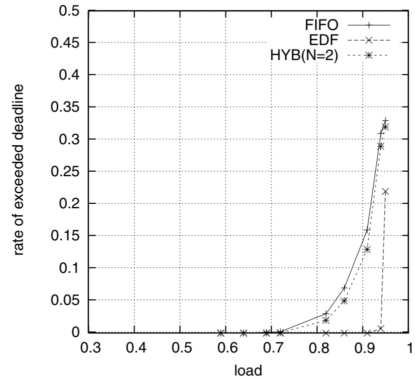

Fig. 2. Comparison EDF, FIFO and HYB (N=2)"Exponentially Distributed ON/OFF Traffic", with deadline set $\mathcal{D}_1$.

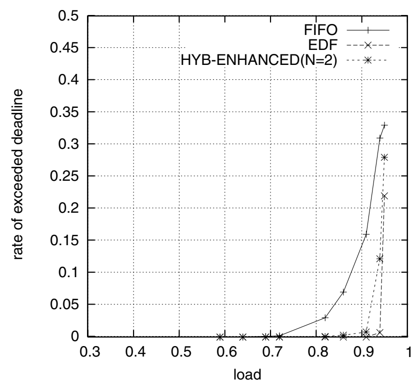

Fig. 3. Comparison EDF, FIFO and HYB-ENHANCED (N=2) "Exponentially Distributed ON/OFF Traffic", with deadline set $\mathcal{D}_1$.

Several traffic types and deadline distributions have been combined in order to investigate the effectiveness of our proposal under various situations.

- For deadlines, we used two sets: $\mathcal{D}_1 = (d_1 = 5ms, d_2 = 50ms, d_3 = 120ms)$, $\mathcal{D}_2 = (d_1 = 10ms, d_2 = 30ms, d_3 = 150ms)$, where d_i stands for the deadline margin assigned to flow i.
- For traffic types, we used ON/OFF bursty traffic (i.e., a sequence of ON and OFF parts). The ON part represents a random period of time during which the source generates traffic, whereas the OFF part designates a random idle period where no traffic is sent. Both the exponentially distributed ON/OFF periods and Pareto distributed ON/OFF periods have been tested.

B. Numerical Results

We compare systematically the performances resulting from different queue cases (pure EDF, pure FIFO, and the 2 variants of the proposed hybrid queue denoted respectively as HYB and HYB-ENHANCED) in terms of the *rate of exceeded deadline*, which gives the percentage of packets whose deadlines are not met. Since the performances of the hybrid queue depends on the value of the parameter N, results for different values of N will be presented.

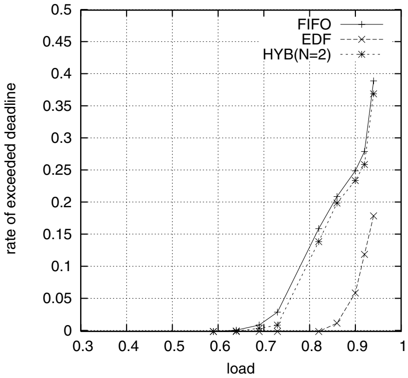

Fig. 4. Comparison EDF, FIFO and HYB (N=2) "Pareto Distributed ON/OFF Traffic", with deadline set $\mathcal{D}_1$.

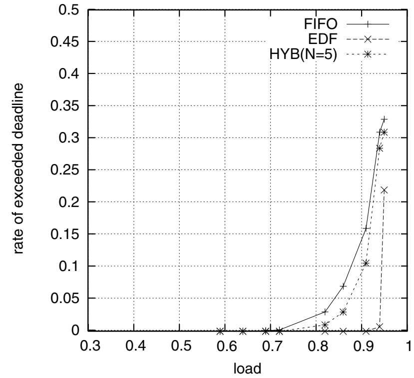

Fig. 6. Comparison EDF, FIFO and HYB (N=5) "Exponentially Distributed ON/OFF Traffic", with deadline set $\mathcal{D}_2$.

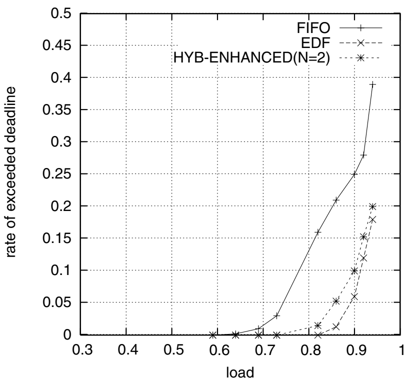

Fig. 5. Comparison EDF, FIFO and HYB-ENHANCED (N=2) "Pareto Distributed ON/OFF Traffic", with deadline set $\mathcal{D}_1$.

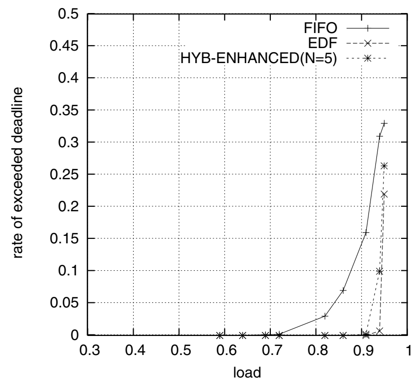

Fig. 7. Comparison EDF, FIFO and HYB-ENHANCED (N=5) "Exponentially Distributed ON/OFF Traffic", , with deadline set $\mathcal{D}_2$.

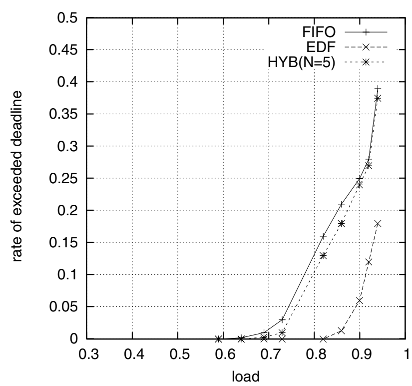

Fig. 8. Comparison EDF, FIFO and HYB (N=5) "Pareto Distributed ON/OFF Traffic", , with deadline set $\mathcal{D}_2$

Figure 2 and Figure 3 provide a comparison between the two variants of our hybrid queue: the *normal mode* and the *enhanced mode*. Both simulations are done with exponentially distributed ON/OFF periods of the ON/OFF traffic, and $\mathcal{D}_1$ deadline set ($\mathcal{D}_1 = (d_1 = 5ms, d_2 = 50ms, d_3 = 120ms)$). We observe that the *normal mode* variant actually has a behavior between the FIFO and the pure-EDF queue (Figure 2). But, it is much closer to the FIFO, compared to the *enhanced mode* which is much closer to pure-EDF (Figure 3). Although this difference is qualitatively forecastable, our simulation results show a big quantitative difference between these two variants.

The same reasoning applies when we consider the case of a Pareto distributed ON/OFF traffic, as illustrated in Figures 4 and 5. We can thus conclude that our proposal is actually effective for two major traffic patterns.

So as already stated, the hybrid queue is able to approach EDF's performances while maintaining a lower complexity level compared with that of the pure EDF queue. This is especially true since we are dealing in this case with a hybrid queue where the scope of the EDF algorithm is limited to its minimum level, i.e. 2 packets ($N = 2$) out of 80.

Based on these results, it is clear that the proposed hybrid queue, especially under the *enhanced mode* of operation, is able to preserve a great deal of the EDF performance advantages while exploiting the FIFO algorithm classification simplicity. The results obtained in case of the first deadlines set $\mathcal{D}_1 = (d_1 = 5ms, d_2 = 50ms, d_3 = 120ms)$ with $N = 2$ are confirmed for the cases where the second set of deadlines $\mathcal{D}_2 = (d_1 = 10ms, d_2 = 30ms, d_3 = 150ms)$ is used with N being increased to 5, as illustrated in Figures 6 to 9.

The number of packets whose deadlines are met increases drastically when N increases. In particular, the performances of

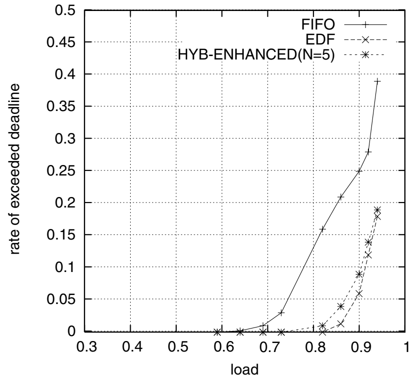

Fig. 9. Comparison EDF, FIFO et HYB-ENHANCED (N=5) "Pareto Distributed ON/OFF Traffic", , with deadline set $\mathcal{D}_2$.

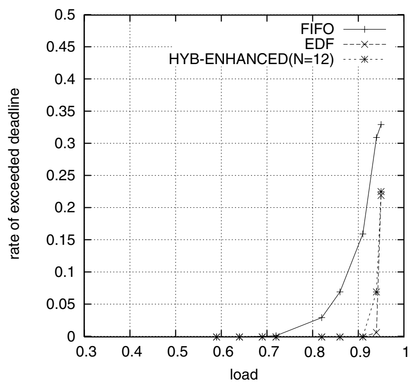

Fig. 10. Comparison EDF, FIFO and HYB-ENHANCED (N=12) "Exponentially Distributed ON/OFF Traffic", , with deadline set $\mathcal{D}_2$.

the so-called *enhanced mode hybrid queue* approaches much more those of the EDF queue (Figure 10). As a result, we are able to reduce drastically the overall complexity through the hybrid queue variants and still maintain reasonable performances (here $N = 12$) as for the number of packets with satisfied deadlines.

IV. Conclusion and Perspectives

We presented in this paper a novel queue management approach that combines two well-known scheduling algorithms, the so-called EDF and FIFO. The main advantage of this hybrid queue resides in reducing the classification complexity of the EDF algorithm, while maintaining to a great extent EDF's advantage regarding packets' deadline respect. We proved this point through a comparative simulation study, where the performances of the proposed hybrid queue were compared with those of pure FIFO, and EDF queues. Our simulation results show first that the hybrid queue outperforms a pure FIFO queue by increasing the number of packets whose deadlines can be met. Furthermore, the percentages of packets with satisfied deadlines resulting from the hybrid queue present a close matching to those obtained by the pure EDF queue.

This validates the interest of the proposed hybrid queue management approach. Analytical modeling of this study is on the way.

References

[1] A.Demers, S.Keshavt, and S.Shenker. Analysis and simulation of fair queuing algorithm. In *Proceeding of ACM SIGCOMM Symposium on Communication Architectures and Protocols*, pages 3–12, September 1989.

[2] A.Parekh and R.Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2:137–150, April 1994.

[3] S.Wang, Y.Wang, and K.Lin. Integrating priority with share in the priority-based weighted fair queuing scheduler for real-time networks. *Journal of Real-Time Systems*, pages 119–149, January 2002.

[4] R.W. Conway, W.L. Maxwell, and L.W. Miller. *Theory of Scheduling*. Addison-Welsey, 1967.

[5] K.Jeffay, D.Stanat, and C.Martel. On non-preemptive scheduling of periodic and sporadic tasks. In *IEEE Real-Time Systems Symposium*, pages 129–139, December 1991.

[6] J.Migge and A.Jean-Marie. Timing analysis of real-time scheduling policies: A trajectoy based model. In *Tech. Report, INRIA, RR-3561*, Nov 1998.

[7] J.Liebeherr, D.Wrege, and D.Ferrari. Exact admission control in networks with bounded delay services. *IEEE/ACM Transactions on Networking*, 4:885–901, December 1996.

[8] V.Firoiu, J.Kurose, and D.Towsley. Efficient admission control for edf schedulers. *Proceedings of IEEE INFOCOM*, pages 310–317, April 1997.

[9] F.Chiussi, M.Gerla, and V.Sivaraman. Traffic shaping for end-to-end delay guarantees with edf scheduling. In *proceedings of International Workshop on Quality of Service (IWQoS)*, June 2000.

[10] D.Ferrari and D.Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected areas in communications*, pages 368–379, April 1990.

[11] L.George, P.Muhletahler, and N.Rivierre. Optimality and non-preemptive real-time scheduling revisited. In *Technical Report 2516, INRIA*, 1995.

[12] L.Georgiadis, R.Guerin, and A.K.Parekh. Optimal multiplexing on a single link: Delay and buffer requirements. In *Proceedings of the IEEE Transactions on Information Theory*, September 1997.

[13] R. Chipalkatti, J. Kurose, and D. Towsley. Scheduling Policies for Real-Time and Non-Real-Time Traffic in a Statistical Multiplexer. In *Proceedings of INFOCOM'89*, pages 774–783 vol.3, April 1989.

[14] T.M. Chen, J. Walrand, and D. Messerschmitt. Dynamic Priority Protocols for Packet Voice. In *IEEE J.SAC.*, pages 632–643 vol.7, June 1989.

[15] J. Peha and F. Tobagi. Evaluation scheduling algorithms for traffic with heterogenous performance objectifs. In *Proceedings of GlobeCom'90*, pages 21–27 vol.1, December 1990.

[16] S. Pingali and J. Kurose. Scheduling Two Classes of Real Time Traffic With Identical Deadlines. In *Proceedings of IEEE GlobeCom'91*, pages 460–465, December 1990.

[17] V. Sivaraman, F. M. Chiussi, and M. Gerla. End-to-end statistical delay service under GPS and EDF scheduling: A comparison study. In *Proceedings of IEEE INFOCOM'01*, pages 1113–1122, April 2001.

[18] A. Grilo, M. Macedo, and M. Nunes. A scheduling algorithm for QoS support in IEEE802.11e networks. In *IEEE Wireless Communications*, pages 36–43 vol.10, June 2003.

[19] M. Andrews. Probabilistic end-to-end delay bounds for earliest deadline first scheduling. In *Proceedings of IEEE INFOCOM'00*, pages 603–612, March 2000.

[20] M. D. Natale and A. Meschi. Scheduling messages with earliest deadline techniques. In *Journal Real-Time Syst.*, pages 255–285 vol.20, 2001.

[21] P.Nain and D.Towsley. Comparaison of hybrid minimum laxity /first-in-first-out scheduling policies for real-time multiprocessors. *IEEE Transactions on Computers*, pages 1271–1278, October 1992.

[22] http: //www.isi.edu/nsnam/ns/.