A hybrid EDF/FIFO queue for efficient real time flow handling

Conference Paper · June 2006

DOI: 10.1109/WFCS.2006.1704145

A Hybrid EDF/FIFO Queue for Efficient Real Time Flow Handling

S. Ould Cheikh El Mehdi, W. Fawaz, K. Chen
L2TI Lab., institut Galilée, University of Paris 13,
99, Av. J.B. Clement, 93 430 Villetaneuse, France
saadbouh@l2ti.univ-paris13.fr, {fawaz, kchen}@galilee.univ-paris13.fr

Abstract

In an attempt to meet the stringent time constraints of real time flows, several scheduling algorithms have been proposed in the literature. Among such algorithms, the so-called EDF algorithm is widely known for its optimal management of flows with strict time constraints, when compared with the standard FIFO algorithm. But still, EDF is complex and expensive as far as implementation is concerned. As a main contribution in this paper, we therefore propose a hybrid EDF/FIFO queue management approach, which consists of a short EDF queue completed by a FIFO queue. This approach allows reducing EDF's implementation complexity while making efficient use of its optimal flow management. Our simulation results underscore the main interest behind this proposal.

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

1. 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, 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, 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], 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 deployment in the network. In this article, our attention will be focused on one of such scheduling techniques, the so-called EDF algorithm (Earliest Deadline First) [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 packet is associated a deadline, which indicates a kind of timely varying priority, or the maximum allowed waiting time. Packets are served according to an increasing order of their associated deadline values. When applied to a networking case ([8],[9]): people propose to associate to each flow , and at the level of a given router along 's path, a local deadline value $^{m}_{i}$. A certain number of theoretical studies ([10],[11] and [12]), 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. However, this algorithm 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 packets; and the remaining packets are scheduled via the simple and fast FIFO algorithm. In the article [13], 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 value. We performed such simulations using the Network Simulator 2 (NS-2) [14], 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.

2. A Novel Queue Management Mechanism

2.1 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 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 [13], 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 , which comprises two serial queues (EDF and FIFO). EDF, as stated previously, handles the first consecutive packets (if any), while FIFO is applied to the $-$ remaining packets. This is the working model of this paper.

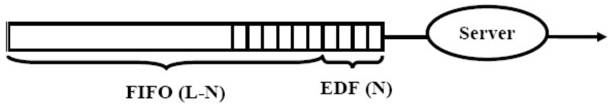

Figure 1. The scheme of our hybrid EDF/FIFO Queue.

It is clear that this hybrid queue aims to achieve a complexity reduction without noticiable loss of performance, when compared to a pure EDF queue. Thus, is a key parameter. The greater the value of will be, the more we will be approaching the performance of a pure EDF queue. In fact, for $=$, 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 will be, the lesser the complexity induced by EDF classification will be. In Section 3, quatitative studies obtained by simulation will be presented.

2.2 Functional description

2.2.1 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), is thus (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 packet 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 packet 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 $(\log_2( ))$ plus the FIFO insertion (either at head or tail) operation (for (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 systemmatic insertion effort. Indeed, the additionnal cost of enhanced mode is $(\log_2)$ 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.

2.2.2 Algorithmic description

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

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

- is the length of the EDF queue, the total length of this hybrid queue;
- $_1$ (rep. $_2$) is the number of packets in the EDF (resp. FIFO queues), initially $_1 = _2 = 0$.

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

New packet insertion Upon arrival of a new packet ($_a$), the insertion is performed either in Normal or Enhanced mode.

- IF (mode=Normal) THEN
  - IF $_1$ THEN insert $_a$ into EDF, $_1 := _1 + 1$
  - ELSE insert $_a$ into FIFO, $_2 := _2 + 1$
- ELSE (in this case mode=Enhanced)
  - IF $_1$ THEN insert $_a$ into EDF, $_1 := _1 + 1$
  - ELSE IF ($_a$ $_l$) THEN move $_l$ (the last packet of the EDF queue) to the head of FIFO, insert $_a$ into EDF, $_2 := _2 + 1$
    - ELSE insert $_a$ into FIFO, $_2 := _2 + 1$
- Check for rejection (of the last one of FIFO), if any :
  - IF $_1 + _2$ THEN Reject the last one of FIFO, $_2 := _2 - 1$

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 $_2 := _2 - 1$,
- ELSE $_1 := _1 - 1$

3. Hybrid Queue Performance Evaluation: Simulation Study

In this section, we evaluate through simulations the performances of the two hybrid queue variants proposed in the previous section, in terms of deadline respect percentage. The main objective is to gauge 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 , the EDF algorithm is applied to the first packets while FIFO is enforced for the remaining packets ($-$ 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 tradeoff between performance improvement and complexity reduction, an appropriate value needs to be assigned for the parameter. Achieving better performances in terms of deadline respect requires increased values of , in which case we approach the performances of an EDF queue. Conversely, reducing the hybrid queue's complexity necessitates a reduced value of the parameter. Building on this analysis, it is clear that is a key parameter in our simulations.

The analytical modelling of this hybrid queue is rather difficult. As a first attempt, we carried out simulation studies using the network simulator 2 package (NS-2) [14], to which we added specific modules implementing the two variants of our hybrid queue. Hereafter, simulation results will be presented for different values of , with a focus on the lowest value of realizing near pure EDF performances.

3.1. Simulation Scenario

In our simulation conducted on NS-2, we consider a network topology comprising 3 traffic sources. These sources share a 2 Mb/s link. As such they share the same queue, which can be either a pure EDF, a pure FIFO, or one of the proposed hybrid queue variants. Moreover, we assume a fixed packet size of 150 bytes

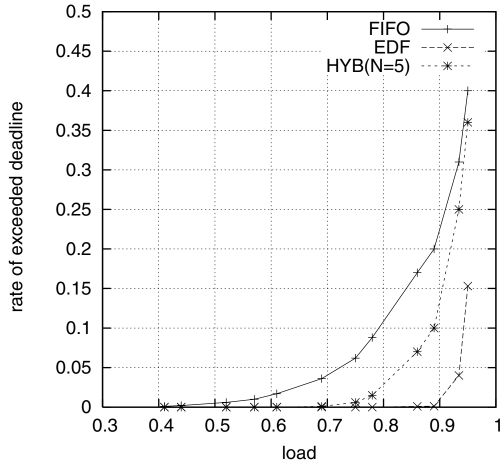

Figure 2. Comparison EDF, FIFO and HYB (N=5)

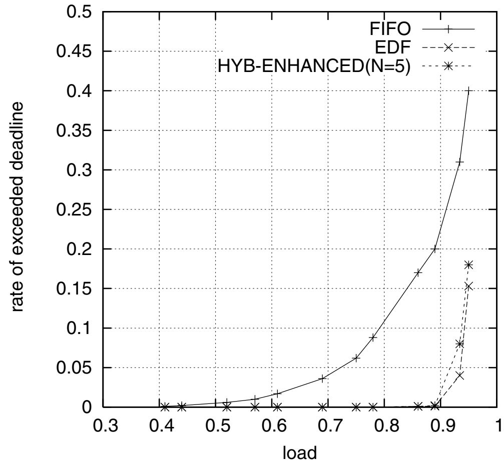

Figure 3. Comparison EDF, FIFO and HYB-ENHANCED (N=5)

for traffic flows generated by the sources. The queue shared by the 3 traffic sources has a total length of 80 (= 80). Finally, the sources generate ON/OFF bursty traffic (i.e., a sequence of ON and OFF parts). The ON part represents an exponentially distributed period of time during which the source generates traffic, whereas the OFF part designates an exponentially distributed idle period where no traffic is sent.

3.2. Numerical Results

We compare the performances resulting from different queue cases (pure EDF, pure FIFO, and the 2 variants of the proposed hybrid queue denoted respectivement 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 , results for different values of will be presented.

Figure 2 reveals the rate of exceeded deadline function of load in the case of different queues implementations, including the proposed normal mode hybrid queue variant (= 5, for the hybrid queue).

Based on this figure, one may observe that the normal mode hybrid queue presents better performances compared with the pure FIFO queue, but slightly degraded performances are obtained relative to the pure EDF queue. This is due mainly to the EDF/FIFO combination of the hybrid queue. Nonetheless, performances degradation relative to the pure EDF queue is improved when adopting the enhanced mode hybrid queue variant as illustrated in Figure 3. Indeed, a near perfect match with the pure EDF performances is observed in such a case.

So, 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 hybrid queues where the scope of the EDF

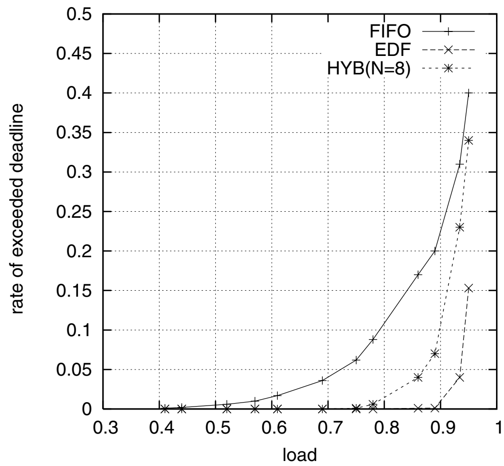

Figure 4. Comparison EDF, FIFO and HYB (N=8)

algorithm is limited to 5 packets (= 5) 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 for = 5 are confirmed for the cases where is increased to 8, or 10 as illustrated in Figures 4 to 7.

The number of packets whose deadlines are met increases drastically in these latter cases (= 8 10), and importantly the performances of the so-called normal mode hybrid queue approaches much more those of the EDF queue (as illustrated in Figures 4 and 6). As a result, we are able to reduce drastically the overall complexity through the hybrid queue variants and still maintain reasonable performances as for the number of packets with satisfied deadlines.

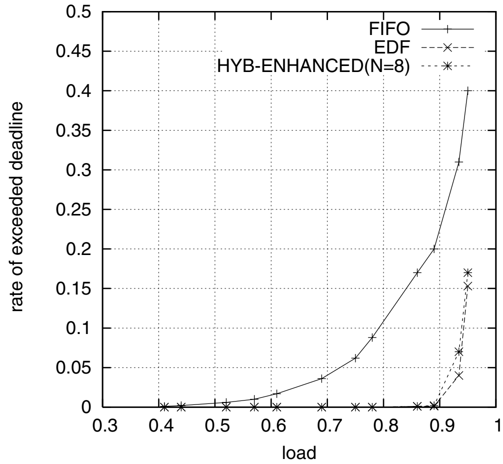

Figure 5. Comparison EDF, FIFO et HYB-ENHANCED (N=8)

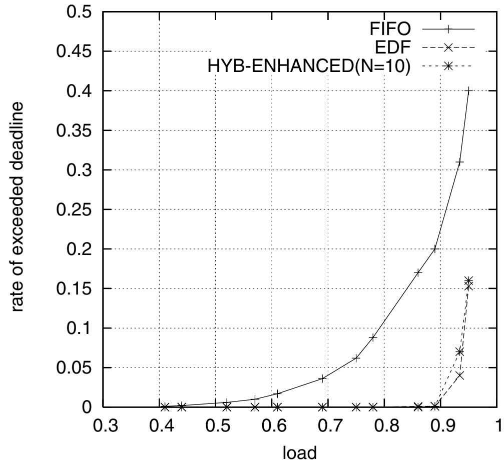

Figure 7. Comparison EDF, FIFO and HYB-ENHANCED (N=10)

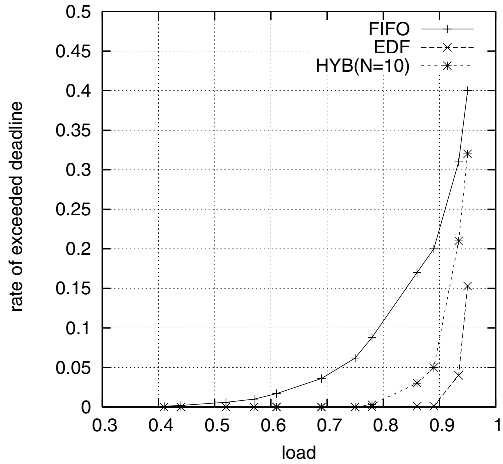

Figure 6. Comparison EDF, FIFO and HYB (N=10)

4. 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 modelling of this study is on the way.

References

[1] S.Keshavt A.Demers 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] Y.Wang S.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] 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.

[9] 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.

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

[11] 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.

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

[13] 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.

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