Zeszyty Naukowe WWSI, No 16, Vol. 11, 2017, pp. 29-41

Simulation studies of EDF scheduler with different deadline distributions

Robert Janowski*
Warsaw School of Computer Science

Andrzej Janowski
Independent scholar

Abstract

In this work we perform extensive simulation studies to evaluate the mean packet waiting time in a system composed of two queues and Earliest Deadline First scheduler with different deadline distributions. The investigation is focused on the analysis of mean packet delays when deterministic and exponentially distributed deadline values are used. Comparing the results is targeted at deciding if the analytical approach proposed in our earlier work for modelling performance of Earliest Deadline First scheduler with exponentially distributed deadline values is also suitable when deadline values are deterministic. The series of simulation tests let us conclude the conditions when both deadline distributions produce very close results thus confirming the applicability of our already published analytical approach.

Keywords – Earliest Deadline First, scheduling, performance evaluation, event simulation.

1. Introduction

In an environment where a number of actors (entities) compete for the same resources there must be a scheme which decides about the order the actors are granted the access to these resources. This scheme is commonly referred to as a scheduling. A number of scheduling algorithms exist each designed to achieve a given purpose.

* E-mail: e-mail: rjanowski@poczta.wwsi.edu.pl

Zgłoszono do druku 13 maja 2017 r.

For example, FIFO (First In First Out) stores all actors in a common queue and serves them in the order of appearance. PQ (Priority Queue) [2] aims to assure a minimum queueing delay for actors served with the highest priority. WFQ (Weighted Fair Queueing) [3] tries to provide a fair resource sharing i.e. the ability to utilize the unused resources among different actors with the guarantees of the minimum amount of resources for each of them. Finally EDF (Earliest Deadline First) [4] was designed to meet some delay requirements imposed for each actor by scheduling for the service an actor with the lowest deadline value. These features have made this algorithm attractive in cases where the delay constraints play the crucial role. In the context of computer or telecommunication networks EDF scheduler is used in 4G mobile networks [5], Wireless Sensor Networks (WSN) [6] or IP networks with real-time traffic to guarantee some delay constraints.

In this paper we have focused on the IP network with actors simply being the packet streams differentiated at a router by a specific value in its header e.g. DSCP (Differentiated Services Code Point) field. We have conducted a series of simulation experiments with two packet streams served according to EDF algorithm and different distribution of deadlines. The simulation studies of EDF scheduler have already been carried out in a number of papers. In [7] the authors studied the performance of EDF scheduler with respect to the average packet delay and the transmission success ratio using commercial simulation software [8]. In [9] the authors performed the simulation tests to analyse the packet delays, throughput, acceptance ratio and the missed deadline ratio with EDF and FIFO schedulers by varying the intensity of the two real-time traffic classes. The simulation studies carried out in [6] were also focused on the evaluation of EDF and Fixed Priority schedulers in terms of acceptance ratio that was defined as the ratio of flows that meet their deadlines to all flows present in the system. The authors of [10] simulated a series of nodes with EDF schedulers and were analysing such metrics as the missed deadline ratio, packet drops, throughput values and packet delays in the environment where the deterministic deadline values applied in each of the nodes sum up to a value imposed as the end-to-end delay requirement.

However to the best of our knowledge no one has simulated EDF scheduler in order to quantify the difference between the values of mean packet delays in case of deterministic and exponentially distributed deadline values. In section 2 we have explained the reasons for such a stated simulation target. In section 3 we have outlined the simulation model and explained few implementation details of our simulator. In section 4 we have provided a number of simulation results that helped us to quantify the investigated difference. In section 5 we have summarised the work and tried to draw some conclusions.

2. Statement of the problem

In [1] we have studied the output port in a packet network node with EDF scheduler and two traffic classes (two packet streams). Packets of each class have been stored in a separate buffer i.e. they have built separate queues. Packets belonging to different classes have been scheduled to the service according to EDF algorithm with exponentially distributed deadline values. An input stream to each class has been modelled as a Poisson stream with intensities λ_1 and λ_2 for class #1 and class #2 respectively. The packets of each class have had a constant length such that their transmission (service) times were S_1 and S_2 for class #1 and class #2 respectively. Such a system with two packet queues and a single service station (modelling the output port) is shown in Figure 1.

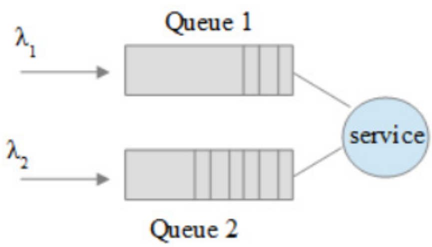

Figure 1. The model of the system with two classes (two queues)

We have observed the system state at the time instants (the epochs) just after the departure of a packet from the service. The system state just after the n-th packet departure we defined as a vector S of 3 random variables: M_n – the number of packets type #1 in the system, N_n – the number of packets type #2 in the system, O_n – the type of packet just served (either 1 or 2 depending on the class of a packet just served). Vector S_n formed a Markov chain [11] i.e. the state of the system at the (n+1)-th epoch could be fully determined based on the information about the system state at the n-th epoch. Exploiting the properties of Markov chains we could then solve the system equations and find out the queue length distribution which let us easily calculate other measures like e.g. the mean packet waiting time which was an important network performance parameter [12],[13].

However we were able to solve the system equations because the size of the state space expressed as a number of all combinations of the allowed values for all 3 random variables forming the vector state S was not so big. It is well known fact that the main obstacles in solving system modelled as a Markov chains it is the size of the state space. In our case the size of the state space was not big because the random variable O_n had only 2 states. It was due to the fact that the deadline values were taken from the exponential distribution and thanks to its memoryless property there was no need to remember previous states but the most recent one i.e. the type of a packet just served (type #1 or #2). However if we have had deterministic deadlines we would have to remember the current deadline value for each packet queued in class #1 or class #2 buffers. This means tremendous growth in the size of the state space from the value $M \cdot N \cdot 2$ (for the case of exponentially distributed deadline values) to the value $(M \cdot N \cdot d_1^M \cdot d_2^N)$ where M, N denote the maximum size of the queues #1 and #2 respectively and d_1, d_2 denote the deadline values of packets from class #1 and #2 respectively.

Such a growth is known in the literature as a state explosion problem and makes the model being not solvable by the above method. In this paper we try to answer the question how accurately the method proposed in [1] might model the system shown in Figure 1 where EDF deadlines have deterministic values. For this purpose we try to quantify the difference in the one of the performance metrics i.e. the mean packet waiting time in case where the deadlines have deterministic or exponentially distributed values. This quantification is carried out by means of the simulation experiments we perform using our own developed tool.

3. Simulation model

We simulate the system as depicted in Figure 1 where the packets of each class arrive to the system according to Poisson process [14] with intensities λ_1 and λ_2 for class #1 and class #2 respectively. The packets of each class have a constant length fixed at the beginning of each simulation test and they are stored in separate buffers i.e. they build separate queues. Packets belonging to different classes are scheduled for the service according to EDF algorithm. Each packet on its arrival is assigned a deadline value that is decreased while a packet is waiting in its queue until it goes into the service. EDF algorithm schedules the packets for the service based on the lowest deadline value. When the currently served packet departs from the system (its transmission ends) and the next packet is to be chosen for the service (i.e. for the transmission) the current deadline values of packets in the head of line position of each queue are compared and a packet with the smallest deadline value is taken to the service. In case one of these queues is empty the packet from the non-empty queue is scheduled for the transmission no matter what its deadline value is (this is the so-called work-conserving property of the scheduler).

For the purpose of the undertaken research as described in section 2 above we have written a program in C++ language that is capable of simulating above defined system with EDF scheduler and both deterministic and random deadline values. In case of deterministic deadline values each packet on its arrival is assigned a deadline being a constant value provided at the beginning of each simulation test. In case of random deadline values each packet is assigned a deadline value drawn from the exponential distribution with parameters provided at the beginning of each simulation test.

The program performs the event simulation i.e. the elapsed time is not changed continuously but from one event to the another. It means that intervals between consecutive events are skipped. Due to this feature the real simulation time (the time-consumption) depends only on the number of simulated events not on the simulation time itself. This makes this simulation model very time effective.

The pseudo-code of the simulation program that implements the above scheduling logic is provided in Figure 2.

The outlined simulation scheme utilizes only four time-related variables that are: the current simulation time (t_current), the time instant of the next class #1 packet arrival (t_arrival_#1), the time instant of the next class #2 packet arrival (t_arrival_#2) and the service completion time (t_service_completion). To perform the simulation, it is sufficient to keep track only of the values of those four variables. If the current simulation time approaches the time of the arrival (either class #1 or class #2) or the service completion time the given event is performed and the time of the next event of the same type is generated.

To be convinced our simulation tool works properly i.e. the components of the simulation model as input traffic generator and EDF scheduler have been implemented correctly we have validated our model. For this purpose, we have performed some simulation tests in so called limit cases i.e. under the conditions the simulator behaves in a special manner that is well known from literature. We have chosen two limit cases: the first one where EDF deadline values are the same for both packet streams and the second one where EDF values are very different. In the first limit case, the system behaves like FIFO scheduler since no packet stream is favoured nor discriminated. In such a case, the performance metrics e.g. mean packet waiting time might be obtained based on the analysis of M/D/1 system [11], [14] and calculated according to the formulae (1). In the second limit case (when EDF deadline values differ very much) the scheduler tends to behave as PQ scheduler. In this case the performance metrics e.g. mean packet waiting time might be obtained based on the analysis of priority queues [2] and calculated according to the formulae (2) and (3) for high and low priority classes respectively.

```
Generate the time instants of the packet class #1 and packet class #2 arrivals (t_arrival_#1 and t_arrival_#2)
Set service completion time equal to the simulation time (t_service_completion = t_simulation)
while (t_current < t_simulation) // the current_time is less than the total simulation_time, continue the simulation
{
        if (t_arrival_#1<t_arrival_#2 and t_arrival_#1<t_service_completion) //the next event is a packet
        class #1 arrival
        {
        update the deadlines for each packet in class #1 queue
        update the deadlines for each packet in class #2 queue
        enqueue the packet in class #1 queue
        increase the class #1 queue size
        assign the current time the value of class #1 arrival instant (t_current=t_arrival_#1)
        generate (schedule) the next class #1 packet arrival (the time instant: t_arrival_#1)
        if(service is empty)
        {
                determine the next packet to be served (based on the lowest deadline value dequeue
                a class #1 or class #2 packet)
                determine the time instant of the packet service completion (class #1 or class#2):
                t_service_completion = t_current+t_serving
        }
}
else if (t_arrival_#2<t_arrival_#1 and t_arrival_#2<t_service_completion) //the next event is a packet class #2 arrival
{
        update the deadlines for each packet in class #1 queue
        update the deadlines for each packet in class #2 queue
        enqueue the packet in class #2 queue
        increase the class #2 queue size
        assign the current time the value of class #2 arrival instant (t_current=t_arrival_#2)
        generate (schedule) the next class #2 packet arrival (the time instant: t_arrival_#2)
        if(service is empty)
        {
                determine the next packet to be served (based on the lowest deadline value dequeue
                a class #1 or class #2 packet)
                determine the time instant of the packet service completion (class #1 or class#2):
                t_service_completion = t_current+t_serving
        }
}
else // the next event is the service completion i.e. t_service_completion <=t_arrival_#1 and t_service_com-
pletion <= t_arrival_#2)
{
        end the service of the packet currently being served (empty the service station)
        update the deadlines for each packet in class #1 queue
        update the deadlines for each packet in class #2 queue
        assign the current time the value of the service completion time (t_current=t_service_completion)
        if (there is a packet in any queue)
        {
                take the next packet to the service (depending on the lowest deadline value dequeue
                a packet from class #1 or class #2 queue)
                determine the time instant of the packet service completion (class #1 or class#2):
                t_service_completion = t_current+t_serving
        }
        else
                set the time instant of the service completion to the value higher than simulation time:
                t_service_completion=t_simulation+1
  }
}
```

Figure 2. Pseudo-code of the simulation program scheduling packets according to EDF algorithm

$$E[W] = \frac{\lambda E[S^2]}{2(1-\rho)} \quad (1)$$

$$E[W_1] = \frac{E[R]}{1-\rho} \quad (2)$$

$$E[W_2] = \frac{E[R]+\rho_1 E[W_1]}{1-\rho_1-\rho_2} \quad \text{where} \quad E[R] = \frac{1}{2}\sum_{k=1}^{2} \lambda_k E[S_k^2] \quad (3)$$

Table 1. The validation of EDF simulator in "limit cases"

Traffic load	Limit case scenario	Theoretical value	Simulated value
ρ_1=0.15 ρ_2=0.15	FIFO (equal deadline values)	EW_1=EW_2=0.214	EW_1=EW_2=0.214±0.002 [1)]
ρ_1=0.15 ρ_2=0.15	PQ (very different deadline values)	EW_1=0.176 EW_2=0.252	EW_1=0.176±0.002 [1)] EW_2=0.252±0.001 [1)]
ρ_1=0.25 ρ_2=0.25	FIFO (equal deadline values)	EW_1=EW_2=0.5	EW_1=EW2=0.499±0.002 [1)]
ρ_1=0.25 ρ_2=0.25	PQ (very different deadline values)	EW_1=0.333 EW_2=0.666	EW_1=0.333±0.002 [1)] EW_2=0.666±0.003 [1)]
ρ_1=0.4 ρ_2=0.4	FIFO (equal deadline values)	EW_1=EW_2=2	EW_1=EW_2=2±0.01 [1)]
ρ_1=0.4 ρ_2=0.4	PQ (very different deadline values)	EW_1=0.666 EW_2=3.333	EW_1=0.665±0.04 [1)] EW_2=3.31±0.02 [1)]
ρ_1=0.48 ρ_2=0.48	FIFO (equal deadline values)	EW_1=EW2=12	EW_1=EW_2=11.8±0.4 [1)]
ρ_1=0.48 ρ_2=0.48	PQ (very different deadline values)	EW_1=0.923 EW_2=23.077	EW_1=0.965±0.02 [1)] EW_2=22.966±0.2 [1)]

[1)] Confidence intervals calculated assuming 95% confidence level.

The analysis of the results gathered in Table 1 leads us to the conclusion that our EDF simulator has been validated favorably.

4. Simulation results

In this section, we present results of two simulation scenarios. The first scenario uses EDF scheduler with two traffic classes and deterministic deadline values while the second one uses EDF scheduler with two traffic classes and exponentially distributed deadline values. By comparing the simulation results from these two scenarios we try to answer the question if the analytical modelling approach outlined in [1] that was proved to be suitable in the case of exponentially distributed deadline values is also appropriate for modelling EDF scheduler with deterministic deadline values. To investigate the subject thoroughly we performed a number of simulation tests verifying the results in different traffic conditions i.e. different traffic loads and deadline values. These simulation results are presented in the following figures (from Figure 3 to Figure 6).

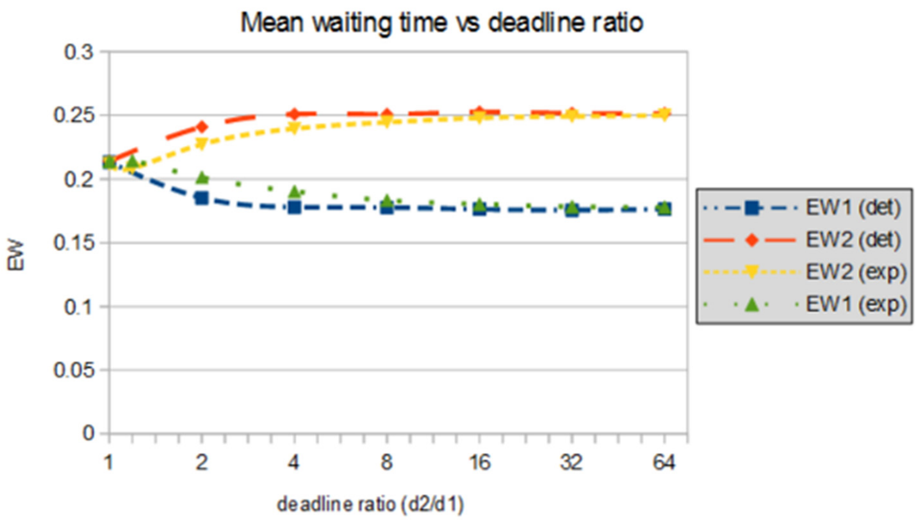

Figure 3. The mean waiting times for both traffic classes (EW_1 , EW_2) and for both simulation scenarios: deterministic (det) and exponentially distributed (exp) deadline values in case of light traffic loads: ρ_1=ρ_2=0.15

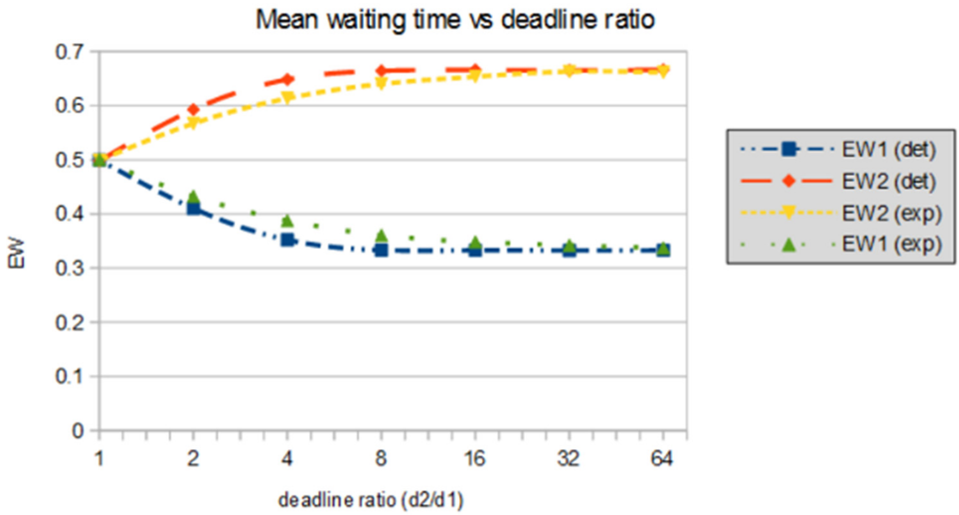

Figure 4. The mean waiting times for both traffic classes (EW_1, EW_2) and for both simulation scenarios: deterministic (det) and exponentially distributed (exp) deadline values in case of moderate traffic loads: $\rho_1=\rho_2=0.25$

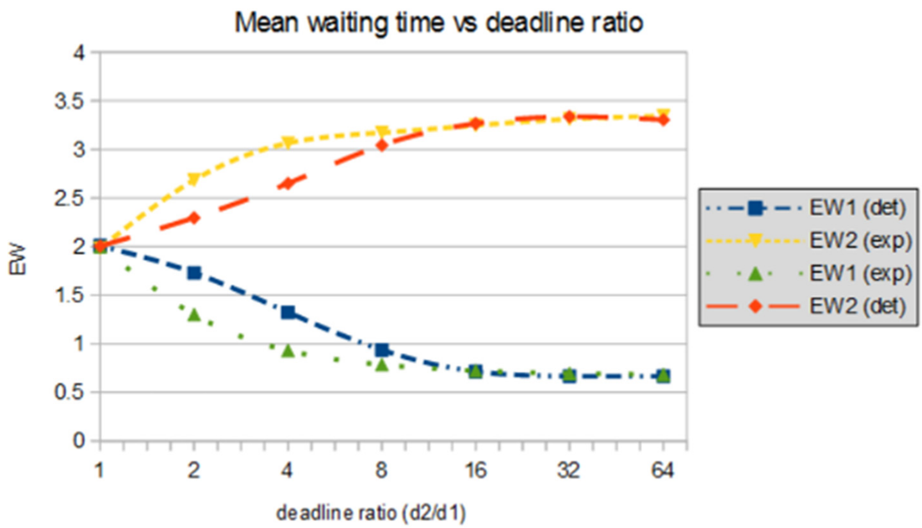

Figure 5. The mean waiting times for both traffic classes (EW_1, EW_2) and for both simulation scenarios: deterministic (det) and exponentially distributed (exp) deadline values in case of high traffic loads: $\rho_1=\rho_2=0.4$

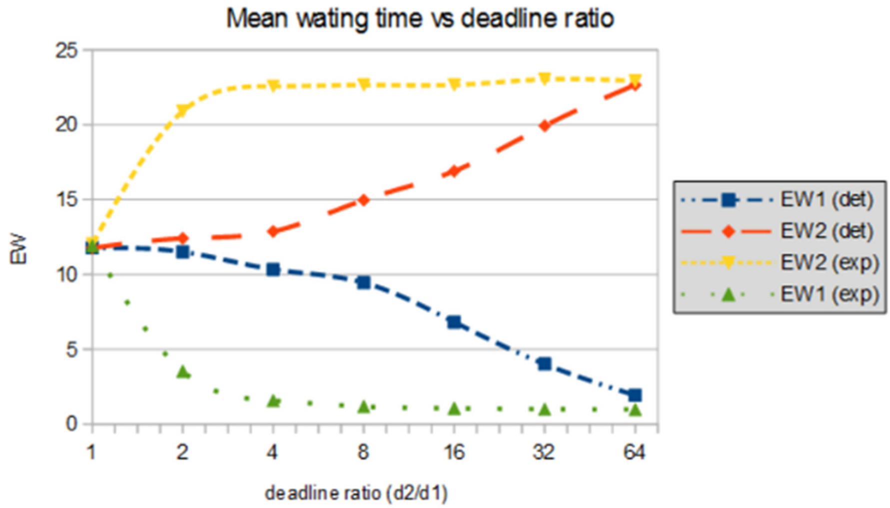

Figure 6. The mean waiting times for both traffic classes (EW_1 , EW_2) and for both simulation scenarios: deterministic (det) and exponentially distributed (exp) deadline values in case of heavy traffic loads: ρ_1=ρ_2=0.48

The presented results have been obtained by repeating each simulation test (for the given traffic load and the deadline value) a number of times and calculating the averages together with confidence intervals. The confidence intervals were calculated assuming 95% confidence level and where enclosed in the above figures although they are not visible since their values are less than 1% of the average value.

The analysis of the results depicted on the above figures provides us with the following observations:

1) The weighted (according to the traffic load of each class) sum of mean packet waiting times of both classes for a given scenario (deterministic or exponentially distributed deadline values) is always constant and equal to the mean packet waiting time of FIFO scheduler with only one queue and the same total traffic load. This is the expected behaviour due to the work-conserving rule of EDF scheduler and it is a one more proof for the credibility of the developed simulator;
2) For light and moderate traffic loads the results for deterministic deadline values provide worse case than results for exponentially distributed deadline values (EW_2 upper-bounds and EW_1 lower-bounds the respective metric values of 'exp' curves);
3) For high and heavy loads the results for exponentially distributed deadline values provide worse case than results for deterministic deadline values (EW_2

upper-bounds and EW_1 lower-bounds the respective metric values of 'det' curves);

4) For high deadline (d_2/d_1) ratio values both curves ('exp' and 'det') converge approaching the results of PQ scheduler. The point of convergence depends on the traffic load i.e. for the case of high traffic load the convergence point is around $d_2/d_1=16$ (see Figure 5) and for the case of heavy traffic load the convergence point is around $d_2/d_1=64$ (see Figure 6);
5) For low deadline (d_2/d_1) ratio values the curves ('exp' and 'det') quickly diverge from the common result of FIFO scheduler.

The above summarized observations we can conclude in the following way:

- The results for FIFO scheduler can be used to analytically model EDF scheduler either with deterministic or exponentially distributed deadline values only in case when d_2/d_1 ratio equals 1. For the d_2/d_1 ratio greater than 1 both curves ('det' and 'exp') quickly diverge that makes FIFO results unsuitable even for very rough approximation of EDF scheduler results;
- The results for PQ scheduler can be used for approximating EDF scheduler performance starting from some boundary value of d_2/d_1 ratio which is dependent on the traffic load;
- Outside the area of applicability of FIFO ($d_2/d_1=1$) and PQ models, the difference between the results for exponentially distributed and deterministic deadline values strongly depends on the traffic load. For the light and moderate traffic loads the differences are not significant. However, as the traffic load grows the results differ more significantly. To express this difference more precisely we introduce the definition of the relative error (4):

$$\delta_x = \frac{\Delta x}{x_0} = \frac{x - x_0}{x_0} \tag{4}$$

where x – is the mean waiting time in case of exponentially distributed deadline values and x_0 is the mean waiting time in case of deterministic deadline values.

Using the definition of the relative error we notice that for the heavy traffic load the difference of the mean waiting times between 'det' and 'exp' scenarios reaches 90% for EW_1 (underestimation) and 75% for EW_2 (overestimation) in the most diverged point. For high traffic, the differences reach 16% for EW_1 and 15% for EW_2. For the remaining traffic loads the differences don't exceed 10%.

5. Summary

In this paper, we have tried to answer the question if we can use the analytical approach described in [1] that proved to be suitable for analysing EDF scheduler with exponentially distributed deadline values also to the same scheduler but with deterministic deadline values. In order to provide the reasonable answer, we have developed our own simulator of EDF scheduler able to work both with deterministic and exponentially distributed deadline values. Using this simulator, we have performed a number of simulation tests focusing on the mean packet waiting time as a main performance metric.

In excessive tests performed for different traffic loads and different deadline values we have found out that except of a light traffic the mean packet waiting time in the case of deterministic versus exponentially distributed deadline values can be underestimated in 90% or overestimated in 75%. This means that applying the analytical method from [1] to the analysis of EDF scheduler with deterministic deadline values and to the derivation of the mean packet waiting time can lead to serious quantification errors. Thus, we can conclude that the analytical method from [1] can be only applicable to the analysis of EDF scheduler with deterministic deadline values in special cases as the light traffic ones. In more general conditions a new analytical method should be developed to provide more accurate results.

Literature

[1] Janowski R., Janowski A. *On EDF scheduler with exponential deadlines*, "Zeszyty Naukowe WWSI" No 15, Vol. 10, 2016

[2] Jaiswal N.K., *Priority queues*, Academic Press, New York, 1968

[3] Parekh A.K., Gallager R.G., *A generalized processor sharing approach to flow control in integrated services networks*, IEEE/ACM Transaction on Networking, Vol. 1. No. 3, June 1993

[4] Stankovic J.A., Spuri M., Ramamritham K., Buttazzo G.C., *EDF scheduling for Real Time systems*, Springer International Series in Engineering and Computer Science, 1998

[5] Capozzi F., Piro G., Grieco L.A., Boggia G., Camarda P., *Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey*, "IEEE Commun. Surveys and Tutorials" Vol. 15, No. 2, Apr., 2013

[6] Wu C., Sha M., Gunatilaka D., Saifullah A., Lu C., Chen Y., *Analysis of EDF Scheduling for Wireless Sensor-Actuator Networks*, ACM/IEEE International Symposium on Quality of Service (IWQoS'14), May 2014

[7] Wu J.C., Hao J., Guo C.Y., *Simulation Research on EDF scheduling algorithm of CAN network based on OPNET*, Advanced Materials Research, Vols. 433-440, pp. 5102-55106, 2012.

[8] OPNET simulation software, http://www.opnet.com

[9] Ighodaro J., *Design and analysis of EDF scheduling algorithm for a chosen network traffic problem: implementation and simulation analysis*, "Journal of Applied Computer Science Methods" Vol. 7, No 1, 2015

[10] Bak S., Cheng A.M.K., Cobb J.A., Leiss E.L., *Load-Balanced Routing and Scheduling for Real-Time Traffic in Packet-Switch Networks*, Proceedings of the 25th Annual IEEE Conference on Local Computer Networks, November 08-10, 2000

[11] Kleinrock L. *Queueing Systems*, Vol. 1-2, Wiley-Interscience, Hoboken, 1975

[12] ITU-T Recommendation Y.1541, *Network performance objectives for IP-based services*, ITU, May 2002

[13] ITU-T Recommendation Y.1540, *Internet Protocol data communication service – IP packet transfer and availability performance parameters*, ITU, January 2002

[14] Gross D., Harris C., *Fundamentals of queueing theory*, 3rd ed., Wiley, Hoboken, 2002