

# Comparison between EDF\_RM and EDF\_DM in dynamic scheduling algorithm with sporadic task

Hemang Thakar

Assistant Professor  
DJMIT engineering college  
Computer Science and Engineering

**Abstract:** Real time system is divided in to two algorithms static and dynamic. Both algorithm want to complete his job in given deadline or before deadline but no one algorithm are successfully complete his job in overloaded and under loaded condition. Dynamic algorithm able to give optimal result in under-loaded condition but not able to give same result in over-loaded condition. Static algorithm not able to give optimal result in under-loaded condition but able to perform well in over-loaded condition. So we think new algorithm to give result optimum in under-loaded condition and give high result in over-loaded condition. We are thinking to mix up dynamic scheduling algorithm EDF (Earlier dead line first) and static scheduling algorithm DM (dead line monotonic). EDF algorithm give optimal result in under-loaded condition and DM algorithm perform well in over-loaded condition. In this paper we compare EDF\_RM and EDF\_DM algorithm. Here EDF\_RM is better than EDF\_DM.

**Keywords:** EDF, LLF, RM, DM, Scheduling Algorithms, Real-Time operating Systems.

## I. Introduction

Real-time systems have well defined, fixed time constraints. Two types of real-time system available hard and soft. Where hard real time system work on principal job must complete their execution before deadline. It's useful where missing deadline may be serious consideration. Soft real time system is more flexible which allow job to execution beyond the deadline. It's applicable where cost of job is more important than can sequences of missing deadline.

### a. Real-Time Scheduler

Generally it's worked in two categories static and dynamic. In case of static algorithm its applied priorities at initial (design) level and remain same throughout the task. In case of dynamic algorithm priority assign on runtime which depended on parameters of task. We can make dynamic scheduling with static priority where Rate Monotonic (RM) and Deadline Monotonic (DM) are two examples of its and Earliest Deadline First (EDF) and Least laxity (Slack) time First (LLF) are belongs to dynamic scheduling with dynamic priority. Here in dynamic priority algorithm are implements in categories first. In which job to change priority daring it is activation and in another type job cannot able to change priority. EDF is job level fixed priority

algorithm of this category. Another side job can change priority during execution called job level dynamic-priority algorithm. Least laxity (slack) time algorithm belongs to it. At time  $t$ , the slack time of a job is  $(d - t)$ , where  $d$  is the jobs deadline and  $t$  is its remaining execution requirement. Here, the slack time is the maximum amount of time a job may be forced to wait if it were to execute on a processor and still meet its deadline. The LSF algorithms give higher priority to jobs with smaller slack time. Since the slack time of a job can change over time, the job priorities can change dynamically.

## II. The Scheduling Algorithm

### a. EDF Scheduling Algorithm

EDF Scheduling Algorithm is called earliest dead line first or nearest deadline first Scheduling Algorithm. EDF is dynamic Scheduling algorithm. The task with the earliest deadline has the highest priority [1]. EDF Scheduling Algorithm gives 100% task utilization when system is in under loaded situation or  $U \leq 1$  But when task utilization cross load factor more then 1 or slightly overloaded processor utilization decreases exponentially [4].

### b. RM Scheduling Algorithm

RM Scheduling Algorithm is called Rate Monotonic Algorithm. RM is a fixed or static priority scheduling algorithm. RM assigns priorities to tasks based on their periods [4]. Disadvantage of this algorithm is that they are not give 100% result in under loaded situation [6]. RM gives better performance in over loaded situation as compare to dynamic scheduling. In RM algorithm shortest period gives first chances to execute but when more than one task have same period then RM randomly selects one for execution next [7].

### c. DM Scheduling Algorithm

DM Scheduling Algorithm is called Deadline Monotonic. DM also fixed or static priority Scheduling Algorithm. DM Scheduling Algorithm is depends on priority. In DM algorithm shortest priority gives first chances to execute but when more than one task have same priority then DM randomly selects one for execution next [2]. In DM Algorithm deadline is greater than execution time. DM is not give 100% result in under loaded situation. But DM gives better performance in overloaded situation as compare dynamic scheduling.

### d. EDF\_RM Scheduling Algorithm

EDF scheduling algorithm perform optimal in under loaded situation and RM scheduling algorithm perform better

performance in overloaded situation as compare EDF. When system is in under loaded then it follows EDF algorithm and system reach slightly overloaded it follow RM algorithm. When two continuous job miss the deadline it will identify as overloaded condition so that this algorithm move on RM algorithm and after 5 job continuous success then the algorithm move on EDF algorithm [4].

**e. EDF\_DM Scheduling Algorithm**

EDF scheduling algorithm perform optimal in under loaded situation and DM scheduling algorithm perform better performance in overloaded situation as compare EDF. When system is in under loaded then it follows EDF algorithm and system reach slightly overloaded it follow DM algorithm. When two continuous job miss the deadline it will identify as overloaded condition so that this algorithm move on DM algorithm and after 5 job continuous success then the algorithm move on EDF algorithm [4].

**III. Simulation Method**

We implement our algorithm in the same environment. The result of propose algorithm are compare with EDF, RM, DM, LLF, EDF and LLF, EDF and MUF in same environment. EDF is a dynamic algorithm and DM is static algorithm. In the periodic task summation of execution time and period of each task [4]. We generated 1 task set from 0.5 to 1.5 and each task set is having 5 periodic tasks. Each task set is simulated for 500 clock cycles. The task is said to be overloaded when load factor is more than 1. Therefore we calculate success full ratio and effective processor utilization [4].

- 1) In the real-time system it is most important thing that every task meet deadline. So we calculate success full ratio as define as [2].

$$SR = \frac{\text{Number of jobs successfully scheduled}}{\text{Total number of jobs arrived}}$$

- 2) Effective processor utilization give information that how many processor are used and we calculate as define as [2].

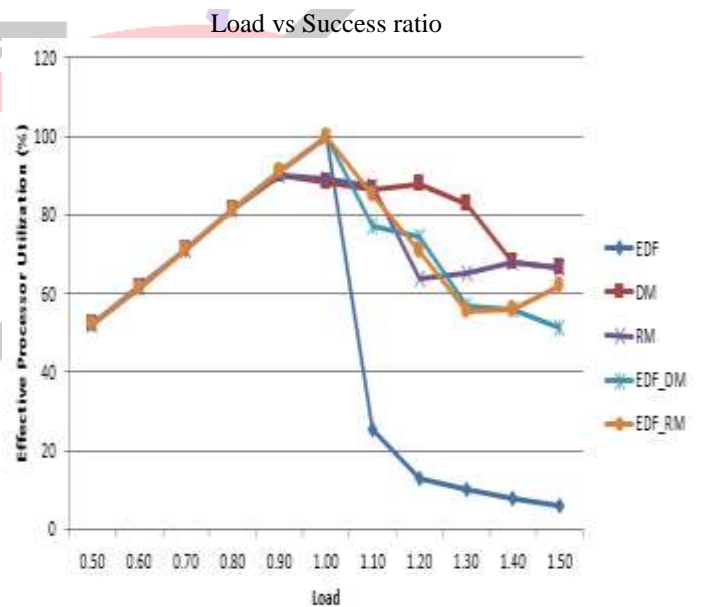
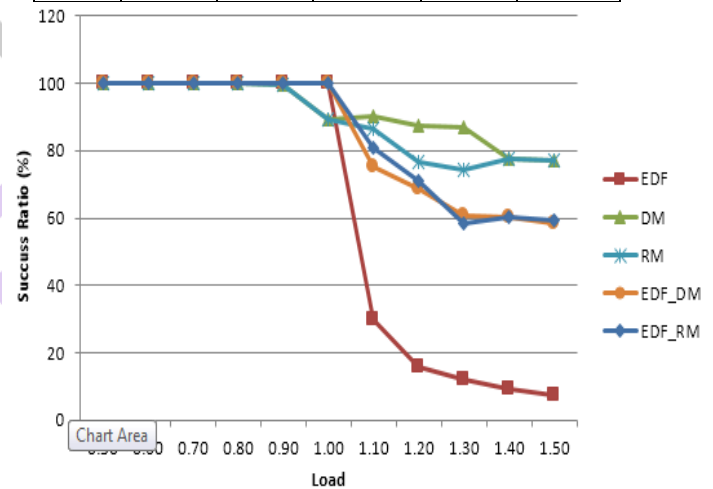
$$EPU = \sum_{i \in R} \frac{V_i}{T}$$

**IV. Final results**

We have taken result for EDF, LLF, RM, DM, EDF\_LL, EDF\_RM, and EDF\_DM

Success Ratio					
LO AD	EDF	RM	EDF_RM	DM	EDF_DM
0.5	100	100	100	100	100
0.6	100	100	100	100	100
0.7	100	100	100	100	100
0.8	100	100	100	100	100
0.9	100	99.5	100	99.3	100
1.0	100	89.4	100	89.3	100
1.1	29.7	86.6	81.3	90.2	75.4
1.2	16.2	76.6	70.8	87.4	68.8

1.3	12	74.3	58.3	86.8	60.7
1.4	9.3	77.3	60.3	77.4	60.3
1.5	7.6	77.2	59.2	77.2	58.2
Effective Processor Utilization					
LO AD	EDF	RM	EDF_RM	DM	EDF_DM
0.5	51.9	51.9	51.98	51.9	51.98
0.6	61.6	61.6	61.62	61.6	61.62
0.7	71.2	71.1	71.2	71.2	71.2
0.8	81.1	81	81.11	81	81.11
0.9	90.7	90	90.78	90	90.78
1.0	100	89	100	88.1	100
1.1	25.3	87.2	84.79	86.5	77.1
1.2	13	63.6	70.87	87.6	74.59
1.3	9.8	65.1	55.46	82.7	56.92
1.4	7.82	68	55.81	68	55.81
1.5	5.87	66.6	61.82	66.7	51.2



Load vs Effective processor utilization

**V. Conclusion**

In this paper we discussed EDF\_RM and EDF\_DM algorithm. Here EDF is dynamic scheduling algorithm with dynamic priority. RM and DM are dynamic scheduling algorithm with static priority. Here EDF\_RM is better than EDF\_DM because EDF\_RM give better result as compare with EDF\_DM.

**References**

- [1] Liu C. L., Layland L., Scheduling algorithms for Multiprogramming in a hard-real time environment, Journal of ACM, Vol 20(1), pp. 46-61, 1973.
- [2] Ketan Kotecha, Apurva Shah, Adaptive Scheduling Algorithm for real-time operating system, in proceedings of IEEE Congress on Evolutionary Computation (CEC 2008), HongKong, pp. 2109-2112, June 2008.
- [3] Ramamritham K., Stankovik J. A., Scheduling Algorithms and Operating Support for Real-Time Systems, Proc. of the IEEE, Vol 82(1), pp. 55-67, 1994.
- [4] Apurva Shah, Ketan Kotecha, Dipti Shah, Dynamic Scheduling for Real-Time Distributed System using ACO, To Appear in International Journal of Intelligent Computing and Cybernetics, (IJICC).
- [5] Locke C. D., Best Effort Decision Making for Real Time Scheduling, Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University, USA, 1986.
- [6] Ramamritham K., Stankovik J. A., Shiah P. F., Efficient Scheduling Algorithms for Real-Time Multiprocessor Systems, IEEE Transaction on Parallel And Distributed Systems, Vol 1(2), pp. 184-194, 1990.
- [7] Carpenter J., Funk S. H., Holman P., Srinivasan A., Anderson J., Baruah S., A categorization of real-time Multiprocessor scheduling problems and algorithms, In Joseph Y.T Leung, editor, Handbook of Scheduling: Algorithms, Models, and Performance Analysis, CRC Press LLC, 2003.
- [8] Hermann Kopetz. Real-Time Systems Design Principles for Distributed Embedded Applications Second Edition
- [9] ALBERT M. K., REAL-TIME SYSTEMS Scheduling, Analysis, and Verification, CHENG, University of Houston