

# Risk Optimization, Resource Partitioning and Mixed Criticality Based Preemptive PP-aware Scheduling for online Service Providers

Madhuri Bedase<sup>1</sup>, D. A. Phalke<sup>2</sup>M. E. Student, Dept. of Computer Engineering, DYPCOE, Savitribai Phule Pune University, Pune, MH, India<sup>1</sup>Assistant Professor, Dept. of Computer Engineering, DYPCOE, Savitribai Phule Pune University, Pune, MH, India<sup>2</sup>

**ABSTRACT:** For real time online service providers, scheduling plays a very important role. The main goal of such application is, to complete the as many as requests before deadline as early as possible and earns profit and reduce the penalty. To achieve this goal, in this paper, system implements the modified version of a profit-penalty aware scheduling (Modified PP-aware). Various features are added to proposed system such as, mixed criticality selection by user to improve the satisfaction, feasibility checking to reduce the penalty, speculated execution order of tasks in ready queue to increase the speed of processing, fair resource allocation by partitioning the interval of Best case (B), Worst case(W) execution time and Deadline(D) intervals and more important threshold risk factor optimization in overload or under load situation. All these factors improve the performance of proposed algorithm to increase the profit and reduce the penalty of online service providers.

**KEYWORDS:** PP-aware, Scheduling algorithm, Optimal risk, Mixed criticality, Resource partitioning, RTOS.

## I. INTRODUCTION

The working of numerous real time applications, is based on the Real Time Operating Systems. For such applications, user request for particular service is an input and as an output these application executes set of requests to provide timely quality of services. The example of such systems includes, traffic control systems, military applications, health industries, online service provides etc. The main challenge is to complete the execution of requests without any delay for buffering. The time required for task execution is measured in terms of milliseconds or nanoseconds. The output of such systems is totally depends on the amount of time taken by processor to process the user requests. The RTOS are divided into three categories based on the timing constraints namely: Hard, Soft and Firm real time systems. Soft RTOS can cross the deadline of task execution. In Hard RTOS if the deadline violation is strictly prohibited as it leads to total failure of system. In Firm RTOS, there is no values earns if deadline constraints not followed.

To manage the execution of all receiving tasks, there is a need of good quality of scheduler in RTOS applications. Some basic scheduling policies are cooperative scheduling, preemptive scheduling, Earliest Deadline First (EDF) approach, etc. The scheduling is very important activity in RTOS applications to provide timely quality of services to their customers. Scheduling is also responsible for the fair resource allocation among all requests to complete them within deadline. Various real time online service providers use the concept of scheduling to provide the on time services to their customers. The Real time online service providers has to provide the services online, to their customers. The Service Level Agreement (SLA) is established between service providers and their multiple customers. In SLA various terms and conditions are written, which should be followed by both service providers and multiple customers. According to SLA, service provider has to provide on time quality services and in return, the person who received the service, has to pay some money to service provider. Such online service providers are also known as time dependent applications. The profit and loss of such system is time dependent that is, if the execution time is increases, profit will reduced and loss will increased. The main aim of such applications is, to complete maximum number of tasks within

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

deadline as soon as possible, thereby make the profit and try to reduce the penalty. To achieve this, there is a need of good quality of scheduler which can achieve the main aim of online service providers.

This paper presents the modified preemptive version of Profit-penalty aware scheduling algorithm (Preemptive PP-aware). The Preemptive PP-aware scheduling algorithm is improved by number of contributions such as mixed criticality approach, partitioning of best and worst case execution time and deadline (D, [B, W]) time intervals, speculated execution order of all incoming accepted task based on two parameters(criticality level and expected gain) and optimized threshold of risk of system to process the tasks. The aim of this proposed system is to speed up the task processing time, minimize the risk of processing task and more importantly to complete the task as soon as possible to earn more profit and minimize the loss/penalty of service providers. All contribution parts are discussed in section III in detail.

The remaining paper includes section II, which shows the survey of related work done in same area. Section III gives the implementation details of system with architecture, problem definition, and overview of system, mathematical model and proposed algorithm. The simulation settings, dataset and comparative results of the system is explore in Section IV. Finally the proposed system is concluded in section V.

## II. RELATED WORK

Scheduling is a very challenging and necessary task in real time system applications. There are numerous scheduling algorithms available. Some of them are explained in [1]. These algorithms are classified based on the execution environment: Multiprocessor or uniprocessor based algorithms. The pre-emptive and non-preemptive Profit and penalty aware scheduling algorithms are proposed in [2]. This scheduling approach is used for time dependent online service providers. Task execution priority is based on highest expected gain. This system with task admission procedure and risk factor was first explained in [3]. The priority free and Discrete Event System (DES) with Supervisory Control Theory (SCT) based model is proposed in [4]. In recent Resource utilization among all tasks of applications is become most challenging task. Shin et al implements an EDF and Largest Weight First (LWF) based scheduling to confirm the execution of task within deadline with minimum utilization of resources in cloud [5]. GlobeAny online scheduling algorithm for cloud service provider proposed in [6] to gain the profit and reduce the penalty. Recent scheduling approaches removes the compulsion that the scheduling analysis is conducted with the same level of criticality or importance. Such types of systems are proposed in [7], [8], [9], and [10]. The problem of fair partitioning of the resources among periodic online tasks in a uniprocessor or multiprocessor environments is solved in [11], [12]. The main aim is to generate the feasible partitioning of all available resources among all tasks for minimum energy consumption without violating time constraints.

## III. PROPOSED METHODOLOGY AND DISCUSSION

### 3.1 System Architecture

Here the application for on-line service providers is presented. We consider the website for any kind of distributors. The proposed algorithm is applied on the service requests of service provider to earn the profit. Fig. 2 depicts the architectural view of proposed system. We consider the file upload application, in which user wants to store their digital data on server. So for this system, input to the system is user request for upload file on server. To process this requests, proposed algorithm that is modified pre-emptive PP-aware scheduling algorithm is used. And finally output is measure in terms of profit or penalty after execution of requests. System incurs profit if the task completes its execution before deadline otherwise system incurs penalty. So main aim of proposed system is to complete the maximum number of task as early as possible within deadline. To achieve this system need to follow the following criteria.

When new tasks are arrives to system, there is need of feasibility checking that whether this task complete within deadline or not. Only feasible tasks are allowed to the system. Pre-emption checking is also necessary task, because frequent pre-emption leads to hanging of system. So we apply the threshold on number of pre-emptions to avoid quick pre-emption. Scheduling point is decides on the time of arrival of new task, abort or dropping of current executing task and pre-empting the task. Task with high criticality level and highest expected gain is executed first. The time at which risk of processing task exceeds the system's tolerable risk, is known as critical point. At this critical point, task should

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

stop its execution even if the maximum amount of execution has completed. Task selection criteria is also important. If we have the prior knowledge of the task's execution order then it is easy to select the next task from ready queue, after finishing current executing task. It will help to reduce the execution time. For this, speculated execution order is used to arrange all tasks in ready queue.

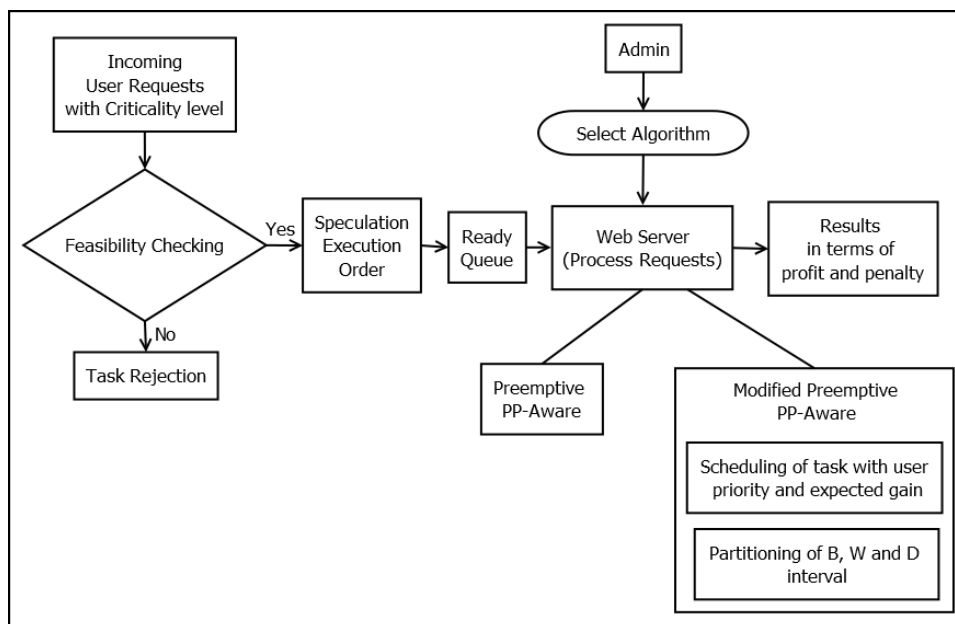


Fig. 1. Architecture of Client Server Application for Online File Upload

As shown in Fig. 1, user can select the criticality level (High, Medium, Low) at the time of requesting to service provider. On arrival of new request, System performs feasibility check to predict the probability that task will finish within deadline. For this purpose task admission test is used. If the task is feasible then it is accepted for further procedure else task is rejected at early stage. All accepted task are then store in ready queue after the speculated sorting for execution, explained in detail later. All tasks are aperiodic, so the ready queue is updated at every time instant according to speculation execution order. At the service provider end, all accepted task in ready queue are scheduled by using either pre-emptive or modified pre-emptive Profit or Penalty Aware (PP-aware) scheduling algorithm. After completion of each task respective profit or penalty is calculated. If the task is completed within deadline system incurs profit, and if the task crossing the deadline or worst case execution time, system incurs penalty. If initially task is accepted and during execution task will aborted, is leads to higher penalty. So main aim of this proposed system is to complete the tasks as soon as possible within worst case execution time to increase the profit and reduce the penalty.

### 3.2 System Overview

**Three Level Mixed Criticality:** Various ideas has been introduced by researchers for analysis of scheduling algorithms. Proposed system allows users to select their urgency of task execution. That is system provides three criticality levels: High, Medium and Low, to the users / customers, which is known as mixed criticality level. This criticality level decides the urgency of task execution for users. There is a highest priority for High criticality level task then medium and finally low.

**Task Admission Test for Feasibility Checking:** On arrival of new task, system has to check the feasibility of processing the task. That is whether this task completes its execution or not within deadline or whether it earns profit or not. To decide this, task admission test is used. In this test, risk factor of each incoming task is calculated based on expected gain, loss and probability of task execution within deadline. There is also a threshold risk factor of system,

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

which is the maximum tolerable risk. If the risk of task is more than threshold, then this task is not accepted for service else task is accepted.

**Speculated Execution Order:** After acceptance, tasks are stored in ready queue. In this queue, all accepted tasks are arranged by using speculated execution order [13]. In this all task are sorted on the basis of two criteria: criticality level of task and expected gain of tasks. The task is high criticality with highest expected gain are at the top of queue, which always execute first at the scheduling point. This speculation execution order, increase the speed of processing tasks, as the tasks are aperiodic in nature, ready queue must be updated at every instance of time.

**Pre-emptive Scheduling:** All tasks in ready queue are scheduled by using pre-emptive or modified pre-emptive PP-aware scheduling algorithm. All tasks are executes pre-emptively. During execution of task, if high priority task arrives, then, this current task execution is pre-empted by upcoming high priority task and that pre-empted task is again store in ready queue as a remaining task  $T_{remain}$ .

**D, B, W interval partitioning:** In existing system, fix range of interval for Best case (B), Worst case execution time (W) and Deadline (D) is used for simulations. For any task, B, W and D are randomly allocated in between [1, 10], [30, 50], [10, 70] respectively. But what happens is that, sometime minimum time span is allocated for large task and vice a versa. Because of this, larger task will not complete in such small time span, to avoid this we make use of interval or resource partitioning concept. In this we divide the intervals according to file size so that system can assign fair interval for each task according to file size to be upload on server.

### 3.3 Mathematical Model

1. Let system S be a web application, applicable for any real time online service providers, where user can upload their digital data on server. The file is an input to the system. That is, the user requesting to server for uploading data on its server. Here, each user request has Variable Execution Time and Known Probability (VEP task). System having such multiple VEP task set for store their digital data on server. So, main aim of the server is to complete all accepted tasks within specified deadline for increasing the profit and reducing the penalty. At the end, output is measured in terms of total accrued profit and total accrued penalty.

$S = \{A, AL, U, C, T, O, P\}$

2. A= Administrator.

Administrator can select the proper algorithm for scheduling all VEP task sets and show results in terms of profit and penalty earn by system.

3. AL= Set of algorithms

$AL = \{AL1, AL2\}$

AL1= Preemptive PP-aware scheduler

AL2= Modified preemptive PP-aware scheduler

4. U= Set of Users

$U = \{u1, u2, \dots, un\}$

U is the set of Users, can upload files on server after successful registration and login.

5. C= Mixed criticality

$C = \{C_H, C_M, C_L\}$

C is the set three criticality, represents tasks processing urgency.

$C_H$  = High criticality

$C_M$  = Medium Criticality

$C_L$  = Low Criticality

6. VEP task set, that is user requests

$T = \{\tau_1, \tau_2, \dots, \tau_n\}$

T is set of VEP tasks, where every task  $\tau_i$  having six parameters such a that,

$$\tau_i = (Ci, Ri, [Bi, Wi], fi(C), Gi(t), Li(t), Di)$$

Where,

C = any one of criticality selected by user.

R= Release time of task

B = Best case execution time allocated for task

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

W = Worst case execution time allocated for task

D = Tasks deadline

$f(C)$  = Variable task execution time with probability density C function

$$f(t) = \frac{1}{(W-B)} \quad (1)$$

$G(t)$  = Completion TUF, represents the obtained gain of task which is completed at t time,

$$G(t) = -ag(t - D), R \leq t \leq D \quad (2)$$

$L(t)$  = Abort TUF, represents the obtained loss of task which is aborted at t time,

$$L(t) = al(t - R), R \leq t \leq D \quad (3)$$

### 7. Process of scheduling all VEP tasks

Pr = {Pr1, Pr2, Pr3, Pr4, Pr5, Pr6}

Pr1 = Task admission test

Pr1 = {e1, e2, e3, e4}

This test is used to take decision regarding to execution of task, aborting a task, and when to abort a task. This test includes following steps.

e1 = Set maximal risk factor  $P_{max}$  of system, which is tolerable risk level of system.

e2 = Calculate Risk Factor (RF) of task when it arrives and at every t instant.

$$RF = \frac{L(t) * P(T > D)}{E(G(T))} \quad (4)$$

Where,

$E(G(T))$  = Tasks expected gain

$$E(G(T)) = \int_{t_s}^{\infty} G(t) \cdot I_{[R,D]}(t) \cdot f(t) \cdot I_{[t_s+B, t_s+W]}(t) \cdot dt \quad (5)$$

$f(t)$  = Variable execution time of task with probability density function

$$f(t) = \left(\frac{1}{B}\right) \cdot I_{[t_0+B, t_0+W]}(t) \quad (6)$$

$I_{a,b}(x)$  = Indicator function

$$I_{a,b}(x) = 1, \text{ if } (x \in [a, b]) \text{ otherwise } 0 \quad (7)$$

$G(t)$  = Completion TUF ...From equation (2)

$L(t)$  = Penalty function ...From equation (3)

$P(T > D)$  = Probability that task cannot be completed before D.

$$P(T > D) = 1 - P(t \leq D) = 1 - \int_0^D f(t) dt \quad (8)$$

e3 = Compare  $P_{max}$  of system and RF of task, if RF is greater than  $P_{max}$ , then task is not accepted or task is aborted during execution, otherwise task is accepted for further execution.

Pr2 = Speculated execution order for task in ready queue

All accepted tasks are arranged in speculation execution order in ready queue and updated at every instance t. The task with highest criticality with highest expected gain is stored at the top of ready queue.

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Pr3= System made scheduling decisions at

- Arrival of new task;
- Successful completion of task execution before deadline;
- When task execution meets previously calculated preemption point, critical time or deadline.

Critical time is the time at which RF is greater than  $P_{max}$ . At this critical time the task is forcefully aborted even if the small amount of execution is remaining.

$$t_{critical} = \inf \{t + t0 : p(t, t0) > P_{max}\} \quad (9)$$

Pr4= Task selection strategy. Highest expected gain with highest criticality task is selected for execution from pending queue. This task is executed till execution reaches to next scheduling point.

Pr5= Apply any one of the three algorithm for scheduling all accepted VEP tasks.

Pr6= Display output in terms of profit and penalty.

$$Total\ accrued\ profit, G(\Gamma) = \sum_{i=1}^n G(\tau_i) \quad (10)$$

$$Total\ accrued\ penalty, L(\Gamma) = \sum_{i=1}^n L(\tau_i) \quad (11)$$

$$Total\ Utility, U(\Gamma) = G(\Gamma) + (-G(\Gamma)) \quad (12)$$

### 3.4 Proposed Algorithm

Input: Incoming Requests from user to upload files on server with selected criticality level

Process:

1. Parameter Initialization { from B, W and D partitioned intervals}
2. Check feasibility
3. If  $Risk_{task} \leq Risk_{system}$
4. Then accept the task
5. Else reject the task
6. For all accepted task, apply speculation execution order
7. Get criticality level and compute expected gain
8. All task rearrange with speculation execution order
9. And store in Ready queue
10. Apply Modified Pre-emptive PP-aware scheduling
11. Select top most task from ready queue for execution
12. Calculate scheduling point  $t_s$
13. Calculate Critical point  $t_{critical}$
14. If true Continue tasks execution
15. Remove Request if following condition satisfy
16. If deadline cross or Risk factor cross
17. Execute currant task if any task is not complete or other task not arrive in same time
18. On arriving new task, check user assigned priority and gain
19. If high, Pre-empt current executing task
20. And remaining task store in pending queue as  $T_{remain}$
21. Process new request as currant request
22. If (currant task complete)
23. Process task which is in waiting state

Output: In terms of total accrued utility, calculated using equation (12)



# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

## IV. EXPERIMENTAL RESULTS

### 4.1 Experimental Setup

For experiment we use, one server and multiple clients. Here we randomly upload the 100 files on server and process it using both pre-emptive and modified pre-emptive PP-aware algorithm. Each task is describe as follows:

$$\tau_i = (C_i, R_i, [B_i, W_i], f_i(C), G_i(t), L_i(t), D_i)$$

Criticality is selected by user at the time of requesting. B, W and D are allocated randomly from following interval based on file size, on arrival of every new task. TABLE 1, represent the partitioning of B, W and D intervals.

The risk factor has great impact on the performance of proposed modified pre-emptive PP-aware scheduling algorithm under overload condition. It is very challenging task to identify the optimal threshold risk factor, which is appropriately working in overload as well as under load situation. In this system, we identify optimal threshold risk factor = 3 for both overload and under load situation. We make this value optimal after conducting multiple experiments with set of 100 tasks.

TABLE I  
B, W, D INTERVAL PATITIONING

File Size	B	W	D
1kb to 10mb	[1,4]	[30, 40]	[10, 25]
10mb to 50 mb	[3, 6]	[35, 45]	[25, 40]
50 mb to 100 mb	[6,10]	[40, 50]	[40, 70]

### 4.2 Dataset

To test the system's performance, randomly generate 100 tasks and upload on server. File might be text file, image file or any other multimedia file. User cannot upload the file, which is greater than 100 mb.

### 4.3 Result Set

The performance of Modified Pre-emptive PP-aware algorithm is compared with existing pre-emptive PP-aware algorithm as shown in Fig. 4. The optimal risk factor and speculated execution order leads to better performance in proposed system. If the system scheduling all tasks using Pre-emptive PP-aware scheduling algorithm, then Profit is 75 % and penalty is 25% percent. But our proposed work increasing the systems profit up to 90 %. This system is better in terms of maximize profit and increasing execution speed also executing maximum number of tasks.

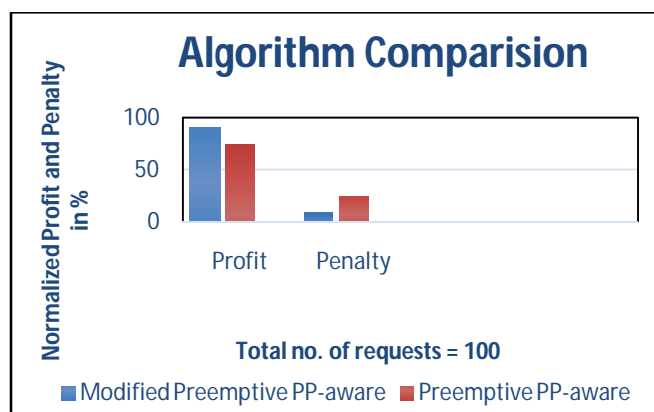


Fig. 2. Comparing proposed algorithm with existing algorithm

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

## V. CONCLUSION

This paper presents the implementation of modified pre-emptive profit-penalty aware scheduling algorithm with mixed criticality and a dividing interval of (D, [B, W]) approach and also with some more contribution like optimal risk of system and speculation execution order of accepted tasks in ready queue. Optimal risk factor has great impact on output whereas speculated execution order increase the speed of processing tasks, which ultimately helps to increase the profit and decrease the penalty of online service providers. Mixed criticality selection improves the user satisfaction and interval portioning leads to fair resource allocation among multiple requests to complete them within deadline. In future this system can be applicable for cloud service providers.

## REFERENCES

- [1] S. Mohammadi, "Scheduling algorithms for real time systems," Queens University, Tech. Rep., 2005.
- [2] S. Li, S. Ren, Y. Yu, X. Wang, L. Wang, and G. Quan, "Profit and penalty aware scheduling for real-time online services," IEEE Transactions on Industrial Informatics, vol. 8, no. 1, pp. 78–89, 2012.
- [3] Y. Yu, S. Ren, N. Chen, and X. Wang, "Profit and penalty aware (pp-aware) scheduling for tasks with variable task execution time," in Proceedings of the 2010 ACM Symposium on Applied Computing, ser. SAC '10. New York, NY, USA: ACM, 2010, pp. 334–339.
- [4] X. Wang, Z. Li, "Optimal Priority-Free Conditionally-Preemptive Real-Time Scheduling of Periodic Tasks Based on DES Supervisory Control," IEEE Transactions on Systems, MAN and Cybernetics: Systems, 2016.
- [5] Shin, SaeMi, Yena Kim, and Sukyoung Lee. "Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing," 12th Annual Consumer Communications and Networking Conference (CCNC), IEEE, 2015.
- [6] Lu, Ping, et al. "Toward online profit-driven scheduling of inter-DC data-transfers for cloud applications," International Conference on Communications (ICC), 2015.
- [7] Burns, Alan, and Robert Davis. "Mixed criticality systems-a review." Department of Computer Science, University of York, Tech. Rep (2013).
- [8] Baruah, Sanjoy K., Alan Burns, and Robert I. Davis. "Response-time analysis for mixed criticality systems." Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd. IEEE, 2011.
- [9] Li, Haohan. Scheduling mixed-criticality real-time systems. Diss. University of North Carolina at Chapel Hill, 2013.
- [10] Succi, Dario, et al. "Multiprocessor scheduling of precedence-constrained mixed-critical jobs," 18th International Symposium on Real-Time Distributed Computing (ISORC), IEEE, 2015.
- [11] Writer's Handbook. Mill Valley, CA: University Science, 1989.
- [12] Aydin, Hakan, and Qi Yang. "Energy-aware partitioning for multiprocessor real-time systems." Proceedings International IEEE Parallel and Distributed Processing Symposium, 2003.
- [13] Liu, Shuo, Gang Quan, and ShangpingRen, "On-line preemptive scheduling of real-time services with profit and penalty", In IEEE Proceedings of Southeastcon, pp. 287-292, 2011.

## BIOGRAPHY

**Madhuri Ravindra Bedase** received the BE in Computer Engineering from S. S. V. P. S. college of engineering which is under North Maharashtra University, Jalgon, in 2011. Currently pursuing ME in Computer engineering from D. Y. Patil College of engineering, Akurdi, Pune, India.

**D. A. Phalke** received the ME in Computer Engineering from D. Y. Patil College of Engineering, Akurdi through University of Pune and she is working as Assistant professor in the Department of Computer Engineering at D. Y. Patil College of Engineering and has 14 year of experience in the field. Currently pursuing PhD in department of technology, Savitribai Phule Pune University.